信 息 安 全 理 论 与 技 术 系 列 丛 书

信任管理与网络安全

蒋文保 著

清华大学出版社

信息安全理论与技术系列丛书

信任管理与网络安全

蒋文保　著

清 华 大 学 出 版 社

北　京

内 容 简 介

本书运用信任管理思想方法，深入探讨了电子商务、移动计算和云计算等所有开放式网络应用环境中共同面临的安全和信任问题。首先系统地介绍信任和信任管理等基本概念和思想方法；其次结合作者多年的科研成果，分析信任管理领域的两个重要研究方向——信任度评估和信任协商的研究内容，重点阐述作者的课题组提出的一些新技术和新方法；最后探讨信任管理思想方法在 P2P 网络安全、网格安全和网络诚信建设 3 个具体问题中的运用。全书共分 8 章，内容包括信任管理概述、基于多维证据的信任度评估模型、基于行为检测的信任度评估技术、自适应自动信任协商模型、自适应信任协商系统设计、信任管理与 P2P 网络安全、信任管理与网格安全以及信任管理与网络诚信建设等。

本书适合网络安全与电子商务相关研究、开发人员阅读，还可以作为计算机及其相关专业研究生和高年级本科生的参考教材，以及培训机构的培训教材。

图书在版编目 CIP 数据

信任管理与网络安全/蒋文保著. —北京：清华大学出版社，2012.11
信息安全理论与技术系列丛书
ISBN 978-7-302-31021-1

Ⅰ. ①信…　Ⅱ. ①蒋…　Ⅲ. ①计算机网络－安全技术　Ⅳ. ①TP393.08

中国版本图书馆 CIP 数据核字(2012)第 304364 号

责任编辑：张　玥　战晓雷
封面设计：傅瑞学
责任校对：白　蕾
责任印制：何　芊

出版发行：清华大学出版社
　　网　　址：http://www.tup.com.cn，http://www.wqbook.com
　　地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
　　社 总 机：010-62770175　　**邮　　购**：010-62786544
　　投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
　　质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
　　课件下载：http://www.tup.com.cn，010-62795954
印 装 者：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：10.25　　**字　　数**：240 千字
版　　次：2012 年 11 月第 1 版　　**印　　次**：2012 年 11 月第 1 次印刷
印　　数：1～3000
定　　价：25.00 元

产品编号：032388-01

前言

目前,基于开放网络环境下的电子商务、移动计算、网络游戏和云计算等新型网络应用正在不断渗入和扩散到国民经济和社会发展的各个领域。在开放的互联网中,由于参与主体数量的规模大、运行环境的异构性、活动目标的动态性以及自主性等特点,各资源主体往往隶属于不同的管理机构,不同的管理域对安全控制的需求和采用的安全策略可能完全不同,使得传统的安全技术和手段,尤其是安全授权机制,如访问控制列表、一些传统的公钥证书体系等,在跨域进行授权及访问控制时显得力不从心,暴露出许多弱点。因此,如何在开放的互联网中建立和维护不同管理域之间以及各个交互主体之间的信任关系,并以此实现它们之间的协同工作,是当前各种新型网络应用所共同面临的一个基础性问题。

1996年,M. Blaze等学者为解决Internet网络服务的安全问题首次使用了"信任管理"的概念,并在此基础上研制了相应的信任管理系统。M. Blaze等人将信任管理定义为采用一种统一的方法描述和解释安全策略、安全凭证以及用于直接授权关键性安全操作的信任关系。信任管理的意义在于提供一个基于安全凭证的通用安全决策支持框架,它比较适合应用于开放的网络环境。与此同时,A. Adul-Rahman等学者从"信任"的概念出发,对信任内容和信任程度进行划分,并从信任的主观性入手给出信任的数学模型用于信任评估。D. Povey在M. Blaze定义的基础上,结合A. Adul-Rahman等人提出的主观信任模型思想,给出了一个更具一般性的信任管理定义,即信任管理是信任意向的获取、评估和实施。目前,许多学者趋于认为信任管理是一种为确定用于决策的信任而通过搜集、分析和编码相关证据以进行决策评价的行为,它实际上是一种决策支持技术。在开放网络环境中,各系统之间相互独立,但只有建立相互信任关系,系统之间才能实现有效交互,所以信任管理作为网络安全技术的重要前提和基础,正日益成为网络安全研究的热点。

在跨域网络协同环境中,由于交互主体间的生疏性以及共享资源的敏感性,陌生的主体之间很难建立信任关系。为了解决上述问题,2000年Winsborough等人提出了"自动信任协商"的概念,它是"通过凭证、访问控制策略的交互披露,资源的请求方和提供方自动地建立信任关系"。目前,自动信任协商的研究已得到迅速发展,并成为当前的一个重要研究方向,其研究和应用在国际上备受关注。

信任管理和信任协商等概念和思想的出现,为所有基于开放、分布、动态特性环境的安全和信任问题提供了新的解决思路。本书运用信任管理思想方法,深入探讨电子商务、移动计算和云计算等所有开放式网络应用环境中共同面临的安全和信任问题。首先,系统地介绍"信任"和"信任管理"等基本概念和思想方法;其次,结合作者多年的科研成果,分析信任管理领域的两个重要研究方向——信任度评估和信任协商的研究内容,重点阐述作者的课

题组提出的一些新技术和新方法；最后，探讨信任管理思想方法在 P2P 网络安全、网格安全和网络诚信建设 3 个具体问题中的运用。

全书共分 8 章。第 1 章“信任管理概述”对信任管理思想方法加以概述。本章对信任管理的思想方法和技术要领进行了简明扼要的阐述，首先介绍信任和信任管理的基本概念，讨论信任管理产生背景、基本内涵、主要内容以及国内外研究现状；然后概要地讨论信任度评估涉及的主要内容，介绍信任协商的基本概念、关键技术和解决方案，以作为本书其他章节的引子。

第 2 章和第 3 章是关于信任度评估方面的研究内容。其中，第 2 章“基于多维证据的信任度评估模型”提出一种基于多维证据的信任度评估模型，该模型基于交易反馈和网络操作行为两个层面的多维证据源进行信任计算，扩展了证据源，突破了只依据单一种类证据源进行信任评估而引起的缺陷；另外，应用改进的 D-S 证据理论来合成多维证据，较好地解决了证据不确定性的问题。第 3 章“基于行为检测的信任度评估技术”专门阐述一种基于网络操作行为的信任度评估模型和算法，这种模型着重考虑了用户的网络操作行为，以用户的日常网络操作行为和交易行为作为信任度评估的依据。

第 4 章和第 5 章是关于信任协商方面的研究内容。其中，第 4 章“自适应自动信任协商模型”提出一种新的信任协商模型，即一种自适应自动信任协商模型（简称 AATN），它能根据信任度评估结果动态调整访问控制策略和协商策略，对于信任度高的协商方可提供快速响应的策略，而对于信任度低的协商方则提供更为谨慎安全的策略，以有效兼顾信任协商中效率和安全两个方面的需求。第 5 章“自适应信任协商系统设计”在第 4 章提出的自适应信任协商模型的基础上，主要讨论自适应信任协商系统的设计和实现问题。

第 6～8 章探讨信任管理思想方法在具体网络安全问题中的运用。其中，第 6 章“信任管理与 P2P 网络安全”主要阐述使用信任协商技术解决 P2P 网络安全问题，重点设计并分析一种 P2P 网络信任协商系统 NetTrust。第 7 章“信任管理与网格安全”在全面分析网格安全需求的基础上，基于信任和信任管理的思想方法深入探讨网格安全解决方案。第 8 章“信任管理与网络诚信建设”探讨在网络诚信建设中如何管理和规范网络主体行为的问题，主要论述软件信任评价体系、网站信任评价体系和网络个人用户信任评价体系 3 种网络主体信任评价体系。

本书作为一本学术专著，来源于科学研究的实践。作者从 2000 年以来一直针对分布式跨域协作环境下的安全问题进行研究，2002 年年底结合网格技术在高能物理领域的应用，完成了中国科学院博士学位论文《网格环境下的安全技术研究》。在此基础上，2005 年 1 月完成清华大学博士后研究报告《开放网络环境下的若干安全问题研究》。在攻读博士学位和进行博士后研究期间，作为主要研究人员参加了中国科学院知识创新工程重大项目之子项目“黑客防范体系”、国家重点基础研究发展规划项目（973）“信息与网络安全体系研究”中的“信息分析和监控”，以及国家 973 项目“信息技术中的应用理论与高性能软件”中的“密码算法研究与网络安全系统研制”等大型项目的研究与开发工作。从 2005 年起任职于北京信息科技大学后，主持承担了北京市教委科技计划项目“基于信任管理的网格安全模型及应用研究”和北京市自然科学基金项目“网格自适应信任协商若干关键问题研究”，2008 年开始主持承担国家自然科学基金项目“一种自适应信任协商模型研究”。作者在梳理这十多年研究经历和研究积累的过程中，深刻认识到信任和信任管理思想方法对于解决开放网络协作环

境下安全问题的重要性，因此下定决心撰写了本书，提供给从事相关技术开发和科学研究的专业人员参考和借鉴。

在研究和写作过程中，本人主要得益于我的研究生杨东浩、刘思征、郭少旭、陈文亮、吴洋、韩莹莹、晁储频、王鸿、汪秋云，本书的很多思想方法是我在指导他们完成科研项目研究和硕士学位论文中产生的，同时他们还帮助我整理书稿，做了许多烦琐的工作。因此，我要对他们表示衷心的感谢！

我还要感谢中国科学院高能物理研究所的杨大鉴老师和许榕生老师，清华大学的戴一奇老师和林闯老师，他们对我在攻读博士学位期间和博士后研究工作中给予了极大的指导和帮助。同时感谢北京信息科技大学各位领导和同事的鼎力相助，感谢清华大学出版社编辑张玥的辛勤工作。

由于时间仓促，加上信任管理是一个较新的研究领域，因此本书错误之处在所难免，欢迎广大读者批评指正。

作者

2012 年 11 月

目录

第 1 章　信任管理概述

目前，基于开放网络环境下的电子商务、云计算、移动计算、网络游戏和物联网等新型网络应用逐渐成为一种主流应用模式。在开放的互联网中，由于参与主体数量的规模大、运行环境的异构性、活动目标的动态性以及自主性等特点，各资源主体往往隶属于不同的管理机构，不同的管理域对安全控制的需求和采用的安全策略可能完全不同，使得传统的安全技术和手段，尤其是安全授权机制，如访问控制列表（ACL）、一些传统的公钥证书体系（PKI）等，在跨域进行授权及访问控制时显得力不从心，暴露出许多弱点。因此，如何在开放的互联网中建立和维护不同管理域之间以及各个交互主体之间的信任关系，并以此实现它们之间的协同工作，是当前各种新型网络应用所共同面临的一个基础性问题。

为了解决上述问题，近年来信息安全领域出现了一个新的重要研究方向——信任管理，目前信任管理技术为所有基于开放、分布、动态特性环境的安全和信任问题提供了新的解决思路。本章对信任管理的思想方法和技术要领进行简明扼要的阐述，以作为本书其他章节的引子。

1.1　信任与信任管理

1.1.1　信任

信任是一个很难严格定义的抽象概念，目前还没有一个精确的、广泛可接受的定义，不同的研究往往倾向于在所处的上下文中对信任这个概念作不同的定义。到目前为止，信任概念在信息技术领域中得到了广泛应用，不同的学者基于不同的目的和角度对信任有不同的定义和理解。

Gambetta[1] 于 1990 年给出了信任的定义，他认为在多代理环境中，信任是在不能提前监控某代理特定行为及影响该行为的相关内容的前提下，在一定程度上某代理能完成该特定行为的主观可能性。随后 K. Konrad[2] 提出信任是一个与诚实、相信、竞争以及可靠等联系在一起的一个主体概念。

Kini 和 Choobineh[3] 在他们的关于信任的理论架构中认为，信任就是：

（1）一个对某一个人或者事情可靠性的预测。其信任度取决一个被信任者的特征、能力、力量以及真实度。

（2）一份对诺言（约定）或者一个关系的条件的责任。

（3）对某一个实体给予信心。

Grandison[4] 认为，信任是在特定的环境中对于一个实体具有诚实、安全和可靠行动的能力的坚定信念。Tyrone[5] 等在 2003 年认为，信任是在特定上下文中，信任者对于被信任者能力、诚实性、安全性和可靠性的量化信念。Mui[6] 则将信任定义为某一代理对另一代理未来行为的主观期待。Wang[7] 认为，信任是某代理根据自身的直接经验对另一节点能力、诚实度和可信赖程度的信念。Hussain[8] 认为，在面向服务的网络环境中，信任是评估代理

对于被评估代理在某既定时间内，按照双方已达成一致的内容，成功地提交该行为的意愿和能力的信念。

总之，信任是一个很广阔的概念，它与忠诚、信赖、安全、可靠和能力等概念都有联系；另外，它在多个学科领域都有涉及，如心理学、社会学、经济学、进化生物学、组织行为学、哲学以及计算机科学等。

文献[9]提出一种方法，将社会学、哲学、管理学、经济学和政治学等领域对信任的研究成果进行划分，从概念上把信任分为六大类。

(1) 倾向信任(disposition)：即人们自然地倾向于信任实体 A。

(2) 环境信任(situation)：即实体 A 信任某一特定的情景(scenario)。

(3) 结构信任(structure)：即实体 A 客观地相信结构 B 是某个整体的一部分。

(4) 信念信任(belief)：即实体 A 相信实体 B 是值得信赖的。

(5) 意图信任(intention)：即实体 A 乐于依靠实体 B。

(6) 行为信任(behaviour)：即实体 A 自愿依靠实体 B。

本书所研究的信任应属于第 4 种信任，即信念信任。我们可以将信任定义为：实体 A 根据相关证据对实体 B 在一定环境下完成特定行为的一种信念。信任具有以下几个主要特征：

(1) 主观性，即信任是一种信念，是一种主观评价。

(2) 非对称性，即实体 A 信任实体 B 并不意味着 B 也信任 A。

(3) 动态性，即实体 A 在经验及所掌握的信息发生变化的情况下对于 B 的信任也会发生变化。

(4) 非完全传递性，即实体 C 向 A 推荐其信任的 B，而 A 根据自己的经验并非一定信任 B。

(5) 环境相关性，即实体 A 在特定环境下信任 B 并不意味着在其他环境下也信任 B。

1.1.2 信任管理

1996 年，M. Blaze 等学者为解决 Internet 网络服务的安全问题首次使用了"信任管理(trust management)"的概念[10]，并在此基础上研制了相应的信任管理系统 PolicyMaker[10] 和 KeyNote[11]。M. Blaze 等人将信任管理定义为采用一种统一的方法描述和解释安全策略、安全凭证以及用于直接授权关键性安全操作的信任关系。基于该定义，信任管理的内容包括制订安全策略、获取安全凭证以及判断安全凭证集是否满足相关的安全策略等。这类信任管理系统的意义在于提供了一个基于安全凭证的、独立于具体应用的、综合的安全决策框架。其本质是使用一种精确的、理性的方式来描述和处理复杂的信任关系。但在这种信任管理思想提出之前和之后，都有一些学者，如 A. Abdul-Rahman 等人认为信任是非理性的[12,13]，是一种经验的体现，不仅要有具体的内容，还应有程度的划分。

A. Abdul-Rahman 等学者从信任的概念出发，对信任内容和信任程度进行划分，并从信任的主观性入手给出信任的数学模型用于信任评估。D. Povey 在 M. Blaze 定义的基础上，结合 A. Abdul-Rahman 等人提出的主观信任模型思想，给出了一个更具一般性的信任管理定义，即信任管理是信任意向的获取、评估和实施[14]。主观信任模型认为，信任是主体对客体特定行为的主观可能性预期，取决于经验，并随着客体行为的结果变化而不断修正。

这类信任模型所关注的内容主要有信任表述、信任度量和信任度评估。这样的一个信任模型与安全策略的实施相结合同样可以构成一个一般意义上的信任管理系统。

我们把上述两种不同类型的信息管理思想分别称为基于凭证的信息管理和基于行为的信任管理,即 M. Blaze 等学者所研究的信任管理模型是一种基于凭证的信任管理模型,而 A. Abdul-Rahman 等人提出的主观信任模型实质上是一种基于行为的信任管理模型。

1. 基于凭证的信任管理

PolicyMaker 是 M. Blaze 等人依据他们所提出的信任管理思想较早实现的信任管理系统,PolicyMaker 为网络服务安全授权提供了一个完整而直接的解决方法,取代了传统的认证和访问控制相结合的做法,并且给出了一个独立于特定应用的一致性证明验证算法,用于服务请求安全凭证和安全策略的匹配。PolicyMaker 是一个实验性质的信任管理系统,其功能相对简单,不提供安全凭证的收集和验证的功能。应用系统必须负责收集并保证足够的安全凭证用于验证相关的操作请求,还需根据安全凭证的公钥信息验证其可靠性,而 PolicyMaker 仅根据应用系统输入的操作请求安全策略集和安全凭证集来完成最后的一致性证明验证工作。这种信任管理引擎与应用系统的功能划分加重了应用系统的负担,而且可能会因为安全凭证收集不充分而导致一致性证明验证的失败。但应用系统负责安全凭证的可靠性验证,使其在选择签名算法时具有一定的灵活性[15]。

KeyNote 是 M. Blaze 等人实现的第 2 个信任管理系统。不同于 PolicyMaker,KeyNote 在设计之初就希望能够促进信任管理系统的标准化并使其易于集成到应用系统中。为此,KeyNote 在系统的设计和实现上与 PolicyMaker 存在着很大的差别。目前,KeyNote 已在 IPsec 协议和网上交易的离线支付等方面进行了一些应用研究。KeyNote 采用一种类似于电子邮件信头的格式来描述安全策略和安全凭证断言。KeyNote 提供一种专门的语言来描述安全策略和安全凭证断言,并且负责安全凭证的可靠性验证。这样一方面减轻了应用系统的负担,使 KeyNote 更容易与应用系统集成;另一方面则有利于安全策略和安全凭证描述格式的标准化,使应用系统能够更有效地传播、获取以及使用安全策略和安全凭证。

REFEREE[16]是 Y. H. Chu 等人为解决 Web 浏览安全问题而开发的信任管理系统。虽然其设计目标比较单一,但可以较完整地实现信任管理模型所列出的各要素。REFEREE 采用了与 PolicyMaker 类似的完全可编程的方式描述安全策略和安全凭证。在 REFEREE 系统中,安全策略和安全凭证均被表达为一段程序,但程序必须采用 REFEREE 约定的格式来描述。REFEREE 灵活的一致性证明验证机制一方面使其具有较强的处理能力,另一方面也导致其实现代价较高。而允许安全策略和安全凭证程序间的自主调用则存在较大的安全隐患。另外,必须看到 REFEREE 的验证结果可能会出现未知的情况。REFEREE 能够在一致性证明验证时自动收集并验证安全凭证的可靠性,应用系统仅需给出初始的安全策略安全凭证和验证内容以及一些必要的验证上下文信息。这一点有利于该信任管理系统的使用。

2. 基于行为的信任管理

1) A. Abdul-Rahman 信任管理模型

A. Abdul-Rahman 信任管理模型是基于社会学特征的信任模型,该模型支持下面的一些社会信任属性:

(1) 信任是依赖于上下文的;

(2) 尽管在一个小的信任值范围内,该模型支持 Agent 诚信的消极和积极信任度;

(3) 信任是基于以前的经验,代理能够识别重复有相似背景和相同代理商的经验;

(4) Agent 通过推荐交换评估信息,从而支持了声誉机制以协助信任决策;

(5) 信任是不可传递的,所有推荐的信任将考虑到推荐的源的信任;

(6) 信任是主观的,不同的观察者对于相同的代理的诚信度可能有不同的看法;

(7) 信任是动态的和非单调的,进一步的经验和推荐能提高或降低另一名代理人的信任程度;

(8) 只有人际信任的支持,在这个阶段中,我们排除倾向性和系统信任。

A. Abdul-Rahman 提出的模型是基于经验和声望的行为信任模型,该模型允许实体决定哪些实体是值得信赖的,并且允许实体调整关于其他实体推荐的知识。该模型可描述为:

(1) 实体直接信任的集合 Q;

(2) 推荐者集合 R;

(3) $C=\{c_1,c_2,\cdots,c_n\}$是实体 x 所知的上下文环境集合;

(4) $A=\{a_1,a_2,\cdots,a_n\}$代表一组实体,直接或者间接与实体 x 进行交易;

(5)经验 e 的结果级别为:$E=\{\mathrm{vg},g,b,\mathrm{vb}\}$。

直接信任评估:直接信任是在一定的上下文中,一个实体信任另一个实体的可信度,用 td 表示。其值的计算取决于直接交易的经验结果:

$$\exists\,\mathrm{td}\in E\,\forall s_e\in s,(s_e=\max(s))\Rightarrow(\mathrm{td}=e); \tag{1.1}$$

其中,s_j 是当经验 $e=j$ 时的经验累加值;如果 max(s)返回的是多个值,那么 td 被分配为不确定的值有如下 3 种情况:当 e 为 vg^g ^? 时,td 为 u+,意为:大部分是好的,小部分是不好的;当 e 为 vb^b ^? 时,td 为 u−,意为:大部分不好,小部分好;当 e 为其他组合的时候,td 为 u0,意为:不好的和好的一样多。

推荐信任:在特定的上下文中 c,一个实体 a 对另一个实体 b 的信任程度 rtd 是由其他实体的推荐信任所给出的。当决定实体 a 在上下文环境是 c 的推荐信任 rtd 时,首先能找到实体 a 在推荐集合 R 里的一个关系(c,a,t),其中 $t=(T_{\mathrm{vg}},T_g,T_b,T_{\mathrm{vb}})$,$T_a=T_{\mathrm{vg}}\cup T_g\cup T_b\cup T_{\mathrm{vb}}$。所以 rtd 的值为

$$\mathrm{rtd}=\mathrm{mod}(\{\forall x\in T^a\,\|\,x\,|\}) \tag{1.2}$$

在 A. Abdul-Rahman 的模型中,除了直接信任和推荐信任的评估,还要进行语义距离的评估和推荐本身的信任评估,评估值公式分别为:

$$\forall e\in E,\mathrm{sd}_e=\mathrm{mod}(T_e) \tag{1.3}$$

$$\mathrm{rd}^*=\mathrm{rd}\oplus\mathrm{sd}_{\mathrm{rd}} \tag{1.4}$$

最终的信任值更新包括了更新经验和结合推荐:

$$s_e=s_e+1 \tag{1.5}$$

$$T_{\mathrm{rd}}=T_{\mathrm{rd}}\cup\{(e\lozenge\mathrm{rd})\} \tag{1.6}$$

$$\forall e\in E\,\forall w_i\in L_e,\quad \mathrm{sum}_e=\sum^{|L_e|}w_i \tag{1.7}$$

其中,w_i 为推荐者的权重,L_e 为那些与 e 的推荐者相关的权重。

2）Beth 信任管理模型

Beth 信任管理模型引入了经验的概念来表述和度量信任关系，并给出了由经验推荐所引出的信任度推导和综合计算公式。在 Beth 信任管理模型中，经验被定义为对某个实体完成某项任务的情况记录。对应于任务的成败，经验被分为肯定经验和否定经验，若实体任务成功则对其肯定经验记数增加，若实体任务失败则否定经验记数增加。模型中的经验可以由推荐获得，而推荐经验的可信度问题同样是信任问题。为此，模型将信任分为直接信任和推荐信任，分别用于描述主体与客体、主体与客体经验推荐者之间的信任关系。即主体对客体的经验既可以直接获得，又可以通过推荐者获得，而推荐者提供的经验同样可以通过其他推荐者获得。直接信任关系和推荐信任关系形成了一条从主体到客体的信任链，而主体对客体行为的主观预期则取决于这些直接的和间接的经验。Beth 信任管理模型所关注的内容主要有信任表述、信任度量和信任度评估。信任度评估是整个信任管理模型的核心，因此信任管理模型也称为信任度评估模型。信任度评估与安全策略的实施相结合同样可以构成一个一般意义上的信任管理系统。P. Herrmann 等人提出了一个“信任适应的安全策略实施”（Trust-adapted Enforcement of Security Policy）的概念，并在这方面做了一些初步的研究。

直接信任定义为“若 P 对 Q 的所有（包括直接的或由推荐获得的）经验均为肯定经验，则 P 对 Q 存在直接信任关系”。当 Q 被信任时，Q 能成功完成任务的概率被用于评价这种信任关系，而概率的计算则取决于 P 对 Q 的肯定经验记录。Beth 采用式(1.8)描述直接信任度与肯定经验记录的关系：

$$V_q(p) = 1 - a^p \tag{1.8}$$

其中，p 是 P 所获得的关于 Q 的肯定经验数，是对 Q 成功完成一次任务的可能性期望。式(1.8)基于 Q 完成一次任务的可能性在[0,1]上服从均匀分布这一假设。

推荐信任定义为“若 P 愿意接受 Q 提供的关于目标实体的经验，则 P 对 Q 存在推荐经验关系”。Beth 采用肯定经验与否定经验相结合的方法描述推荐信任度。推荐信任度与经验记录的关系采用如下公式描述：

$$v_r(p,n) = \begin{cases} 1 - a^{p-n} & p > n \\ 0 & \end{cases} \tag{1.9}$$

其中，p 和 n 分别是 P 所获得的关于 Q 的肯定经验和否定经验数。

在 Beth 信任管理模型中，经验可以通过推荐获得。而对于同一个信任关系，多个不同的经验推荐者可能形成多条不同的推荐路径。这就需要有一个计算方法能够推导并综合所有推荐路径的经验信息，以获得一致的信任度。Beth 分别对直接信任和推荐信任进行了讨论，并给出了相应的信任度推导和综合计算公式。假设 A 对 B 的推荐信任度为 V_1，B 对 C 的直接信任度为 V_2，B 对 D 的推荐信任度为 V_3，则 A 对 C 的直接信任度推导公式表述为

$$V_1 \odot V_2 = 1 - (1 - V_2)^{V_1} \tag{1.10}$$

A 对 D 的推荐信任度可以简单地表述为 $V_1 \times V_3$。Beth 模型还给出了推荐信任度综合计算公式：

$$V_{com} = \frac{1}{n}\sum_{i=1}^{n} V_i \tag{1.11}$$

其中，V_i 是由单个推荐路径推导出的信任度，综合推荐信任度 V_{com} 是这些单个信任度的简

单算术平均。设 $P_i(i=1,2,\cdots,m)$是推荐路径上各不相同的最终推荐实体，V_{ij} 表示其最终推荐实体为 P 的各条推荐路径的信任度，则直接信任度综合计算公式表述为

$$V_{\text{com}} = 1 - \prod_{i=1}^{m} n_i \quad \prod_{j=1}^{n_i} (1 - V_{i,j}$$

算;推荐主要用于信任度的推导计算。详细的主观逻辑算子的描述参见文献[36]。

与 Beth 模型相比,Jøsang 模型对信任的定义较宽松,同时使用了观念空间中的肯定事件和否定事件对信任关系进行度量。模型没有明确区分直接信任和推荐信任,但提供了推荐算子用于信任度的推导。Jøsang 模型的信任度使用三元组来表示,而不是 Beth 模型中的单一数值。但 Jøsang 模型同样无法有效地消除恶意推荐带来的影响。另外,在 Jøsang 信任管理模型中提供了一套主观逻辑算子,用于信任度之间的运算,因此 Jøsang 模型实际上也认为信任的主观性和不确定性与随机性是等同的。

1.2　信任度评估

如前所述,主观信任管理所关注的内容主要有信任的表述、信任度量和信任度评估,其中信任度评估是核心。信任度评估定义了信任关系的量化表示方法、操作、信任关系的传递途径和计算方法。信任度评估采用一种相对的方法对安全信息进行度量和评估,其直接目的是为信任决策提供支持以确立信任关系,与安全策略的实施相结合可以构成一个一般意义上的信任管理系统。信任度评估可以抽象地理解为用一个或者一组算法对影响主体信任度的相关证据进行处理并获得信任度的过程。

1.2.1　信任度评估证据

目前信任度评估所依据的证据通常有两类,即凭证证据和行为证据。凭证证据是指网络主体所拥有的某些数字凭证。基于凭证证据的信任评估通常运用在安全授权、访问控制及信任协商等系统中,这类信任度评估的基本思想是:如果主体 B 拥有并提供主体 A 所要求的相关数字凭证,那么 A 就信任 B。行为证据是指可证明网络主体的网络行为的事实或者记录。相对于基于凭证证据的信任度评估,有人将基于行为证据的信任度评估称为主观信任度评估,这类信任度评估的主要思想是通过考察主体过去的行为来判断是否信任该主体。

在进行信任度评估时需要对证据进行处理,其基本方法是根据原始证据提取出影响目标主体信任度的因子(本书称之为信任因子),并根据需要选取一组影响因子进行信任度评估。

现有信任度评估模型中最常出现的信任因子主要有交易额、交易量、交易时间、交易结果和交易评价等。

(1) 交易额和交易量。交易额和交易量是对货币与货物的计量,通常情况下人们对于交易额和交易量较高的交易会比较重视,所以交易额和交易量较高的证据在信任评估中对于信任度的影响程度也应该较高。因此,很多学者将交易额和交易量作为信任因子引入自己的信任评估模型中。另外,网络上的交易行为不仅限于货物与货币的交换,在其他交易活动中,交易额和交易量可演化为其他标识交易重要程度的变量,比如在网格计算中演化为运算量或者数据交换总流量等。

(2) 交易时间。随着时间的推移,主体在某一时刻的行为对于主体信任的支持力度将会衰减,这与人类社会中人与人之间的信任关系的特点相一致。通常情况下,对于多年未见并失去联系的老友,我们可能很难像以前那样信任他了。因此在很多信任评估模型中,时间

被学者们充分重视。比如,1994 年,Marsh[18] 在其提出的信任模型中首次强调了时间也是计算信任值的一个关键变量;2006 年常俊胜等人[19] 提出了一种基于时间帧的动态信任模型,在这个模型中时间因子的重要性被空前重视。

(3) 交易结果和交易评价。交易结果和交易评价都是表现交易质量的信任因子,是业务反馈证据的核心内容,是信任评估的直接依据。二者有关联,但不可相互替代,交易结果考察当前交易是否成功完成,而交易评价是对成功或者失败的交易给出的意见,交易结果固然直接影响交易评价,但是成功的交易并不一定获得积极的评价,反之亦然。比如,在一次没能最终完成的交易中,客户可能因为店主热心的服务而给出积极的评价;反之,买到商品的客户可能因为某些原因给出消极的评价。

在很多信任度评估模型中提到了直接信任和间接信任的概念,但是不管是直接信任还是间接信任,其根本的信任影响因子无外乎以上 3 类。现有的信任评估模型几乎都是基于以上 3 类信任因子进行信任度评估,不同的是在不同的模型中可能包含的信任因子有所不同。

1.2.2 信任度评估算法设计

算法是信任度评估的又一核心内容,其任务是对获得的证据用数学方法进行处理并计算信任度。无论哪一个信任度评估模型都有一个或者一组算法。经过大量的调查研究,本书将现有主要的信任度评估算法总结如下。

1. 简单加和法或加权平均法

简单加和法是指对选取的信任因子进行量化并采用求和的方法来计算信任值。这类信任度评估模型选取交易评价作为信任影响因子,分别计算正面评价和负面评价各自的总数目,然后用正面评价的数目减去负面评的数目作为主体的信任度。eBay、Amazon 和淘宝等电子商务网站的名声系统使用的就是简单加和法。此外,也有学者将正面评价占总评价数的比例作为主体的信任度,因为这类方法在计算正面评价和负面评价的数目时也是采用简单加和的方法,所以本书将之归类为简单加和法。

加权法本质上与简单加和法一致,不同点在于,加权法区别对待每一条证据,以一定的标准将证据分类并赋予不同的权重,最后基于这些权重对证据数据进行合成。不同的信任度模型对于证据的分类方法以及权重的设置方法可能大不相同。比如,Abdul-Rahman 等人[17]将交易记录分为直接经验和推荐经验,把通过直接经验计算的信任值称为信任,把通过推荐经验获得的信任值称为声望,再用加权的方法把这两个值合并成信任度。

2. 贝叶斯法

贝叶斯法是指利用贝叶斯网络(Bayesian network)来描述不同的信任因子,并将这些因子合成起来以获得信任值。贝叶斯网络亦称信念网络(belief network),它是一种模拟人类因果推理思维过程的模型。贝叶斯法为信任度评估提供了更为合理的理论基础,但是该算法相对复杂且难于理解。

A. Jøsang 在其提出的基于贝叶斯网络的信任模型[20]中,将正面和负面的评价作为输入,并通过 beta 概率密度函数的统计更新来计算信任度。在文献[21]中,Wang 等人提出了利用贝叶斯网络来描述影响信任度的不同因子,并将这些因子合成起来。该文献中选取的

信任因子包括用户对自己节点的信任、对服务提供者的信任、对提供推荐的节点的信任以及对所在团体的信任等。陈建刚等人[22]以网格应用为前提提出了一个基于贝叶斯网络的信任模型，该模型依赖于在选取的信任链路上的所有节点处获得的目标节点的相关属性，合并这些属性以计算信任值。该模型中，信任链路的选择以及最终合并相关属性均基于贝叶斯方法。

3. 模糊逻辑推理法

基于模糊逻辑（fuzzy logic）的信任模型使用语言上的模糊概念来表示信任和信誉，使用隶属函数来描述主体的信任等级，并将模糊逻辑作为模糊值的推理规则。

Song 等人[23]提出的 PowerTrust 系统就是一个基于模糊逻辑推理的信任评估系统。虽然 PowerTrust 在最后计算信任值的时候使用的是加权平均法，但是权重的分配是通过模糊方法计算获得的，即由节点的信用值、交易的时间和交易的数量这 3 个变量的模糊值来确定权重。Ramchurn 等人[24]利用模糊集来指引网络主体对过去的交易进行评价并重新评估彼此之间的信任关系。该模型基于置信度和信誉值，其中置信度通过分析某个主体的交易历史来获取，信誉值通过分析从网络中其他主体处获取的经验获得。国内学者唐文等人[25]也提出了基于模糊集合理论的信任评估模型，该模型提出了一种信任的度量机制，运用模糊 IF-THEN 规则，对人类信任推理的一般知识和经验进行了建模，提出了一种灵活直观、具有很强描述能力的形式化的信任推理机制。

4. 基于 D-S 证据理论的证据合成法

D-S 证据理论（evidence theory）是与概率论相关的理论框架，但所有可能结果的概率和没必要一定为 1，剩余的概率可以看作是不确定性。D-S 证据理论采用信任函数作为度量，不必给出精确的难以获取的概率，可以不需要给出先验概率和条件概率密度，具有处理随机性和模糊性所导致的不确定性的双重功能；D-S 证据理论将“不确定”和“不知道”区分开来，符合人类思维习惯[26]。基于 D-S 证据理论的信任度评估模型已经在 1.2 节中做了详细介绍，此处不再赘述。

除了以上几类算法外，还有其他信任算法，如利用代数符号或图形等表达信任产生和传递过程的算法，基于熵理论、半环代数理论和博弈论等的方法，因为基于这些方法研究信任度评估的成果不多，所以不再一一介绍。

1.2.3 信任度评估模型分类

依据部署模式可以将基于行为证据的信任度评估模型分为集中式信任度评估模型和分布式信任度评估模型两种。不同的部署模式直接影响信任评估参与者之间的数据交换模式以及整个信任评估系统的数据传输、数据存储和信任计算等工作模式。

1. 集中式信任度评估模型

集中式信任度评估模型基于固定的处理中心实现信任评估，我们称这个固定的处理中心为信任中心。信任中心采用集中的方式获得信任信息，它自身可以获得全局信息（比如交易中心对交易的记录）或者通过收集用户的反馈信息实现信任评估（比如电子市场的信誉系统）。

集中式信任度评估系统原理如图 1.1 所示。在集中式信任度评估系统中，存在一个或

者少数几个中心实体负责收集网络参与实体的历史交易记录信息，以这些记录为证据进行信任度评估，并管理和发布网络中实体的信任度。在一定的网络应用环境中，其他参与者可以向信任中心请求相关主体的信任信息，并根据信任度进行交易决策。通常情况下，拥有高信任度的主体可以获得更多的交易机会。

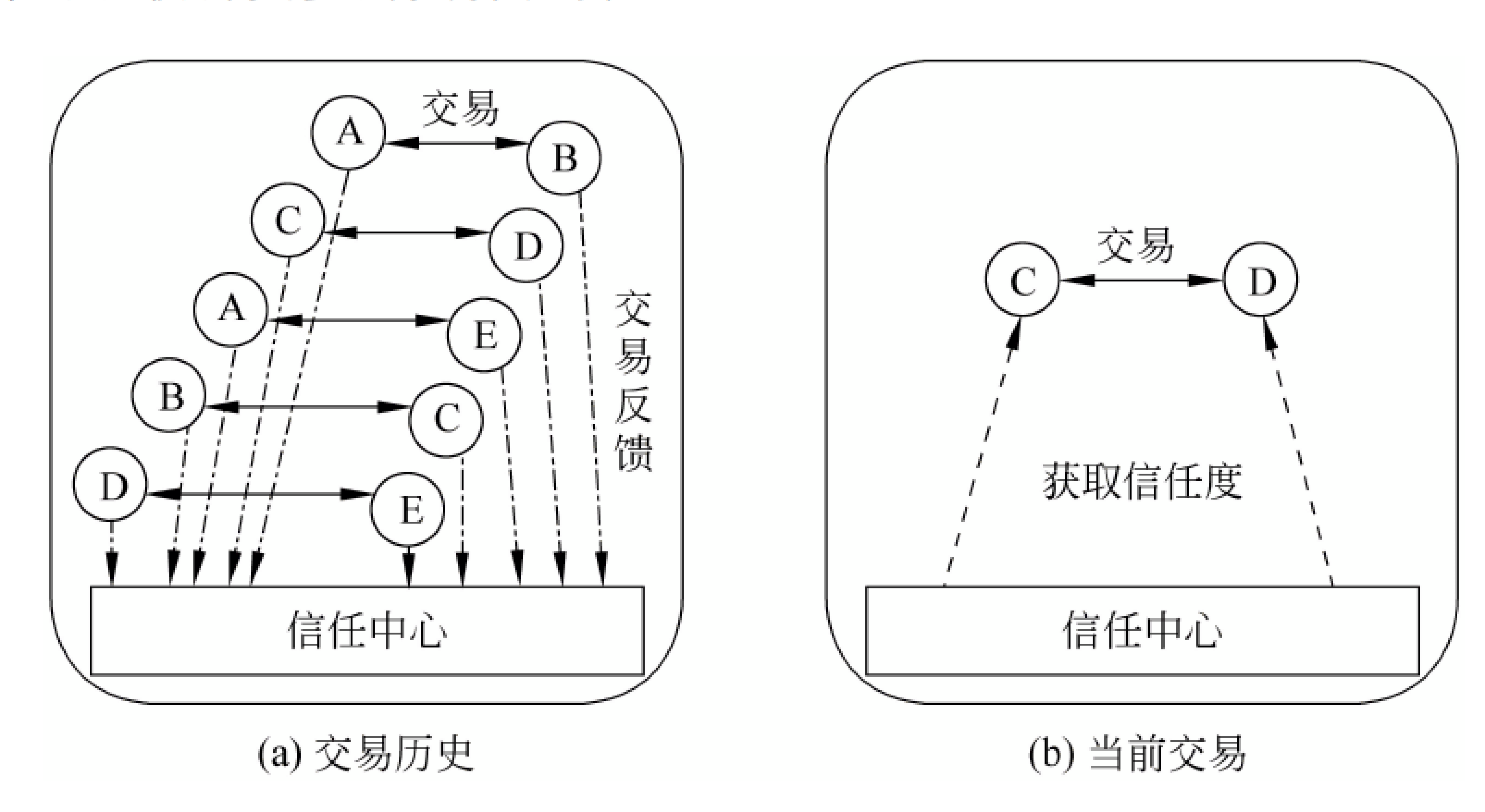


图 1.1　集中式信任度评估系统原理简图

早期的信任模型大多采用集中式部署模式，比较著名的集中式信任度评估模型有 Marsh 的信任模型、Y. Teng 等人基于 D-S 理论的信任模型、Ginsberg 的模型和 Manchala 提出的基于模糊逻辑的信任模型等。目前基于集中式的信任系统广泛应用于电子商务领域，比如著名的 eBay、Amazon 和淘宝等网站采用的在线信誉评估系统都属于此类信任度评估模型。

集中式信任度评估模型的主要特点是模型比较简单，易于实现，但同时也存在一些问题，比如这一类模型对信任中心的效率性、可靠性和可信性的要求都相当高，然而信任中心又必然是最繁忙、最容易成为攻击目标的节点。

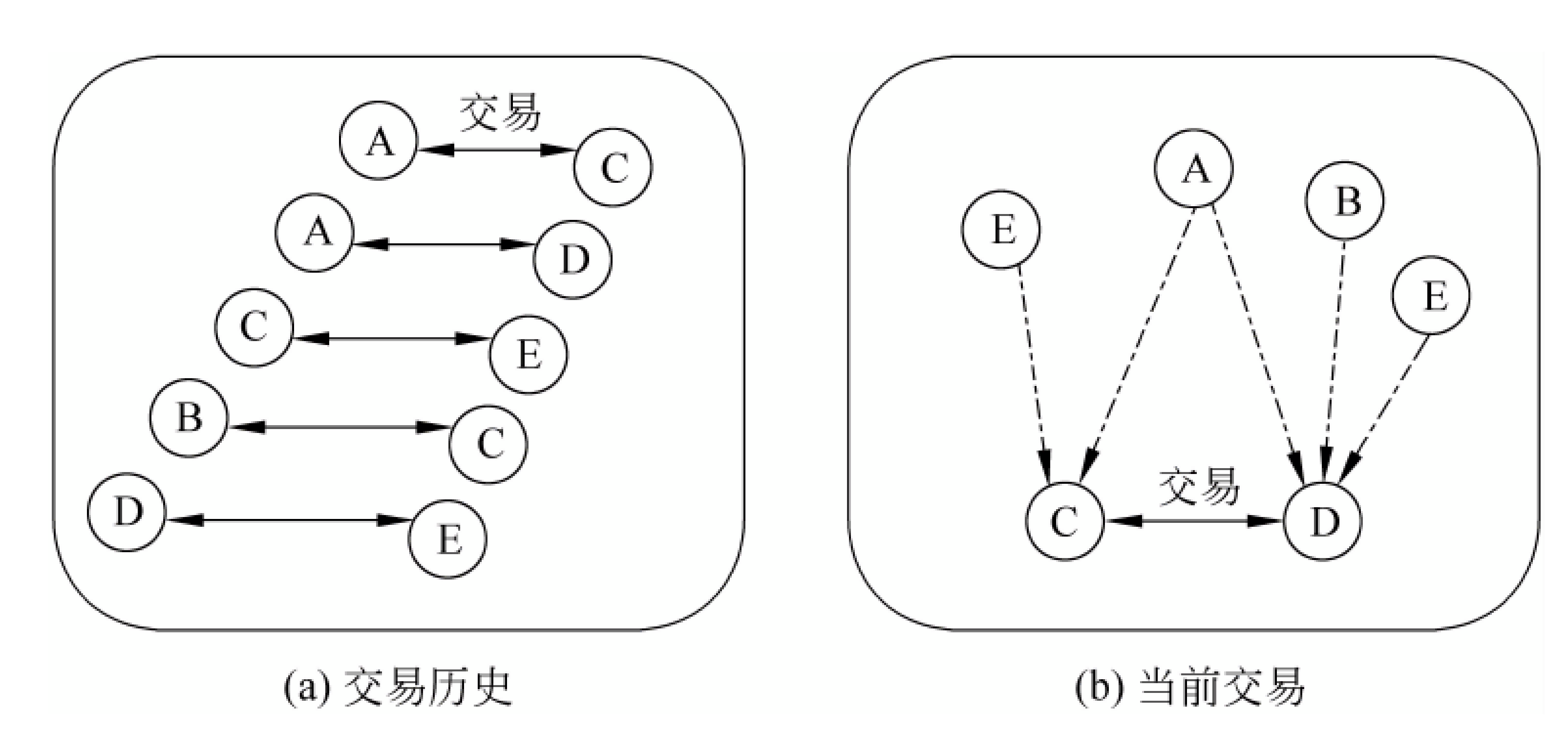
2. 分布式信任度评估模型

分布式信任度模型中不存在类似于集中式信任度评估模型中的信任中心，信任度评估的证据搜集、处理及信任度的计算和管理以分布式的方式由网络实体自身来完成。分布式信任模型更加类似于人类社会的信任建立方式，评估所依赖的证据依然是交易反馈证据，评估算法也无外乎前面介绍的那些算法。

分布式信任度评估系统原理如图 1.2 所示，关于历史交易的记录由各个网络主体自己保存，当需要进行新的交易时，网络主体 C 根据本地存储的关于 D 的交易记录和通过其他主体获得的关于 D 的交易记录来评估 D 的信任度，主体 D 以同样的方式对 C 做信任度评估，在信任度评估的基础上双方建立信任关系。

在 P2P、网格计算、移动自动组网、普适计算和传感器网络等应用中，分布式信任度评估模型得到了充分重视，也逐渐成为研究热点。比较著名的分布式信任度评估模型主要有 T. Beth 等人提出的信任度评估模型、Abdul-Rahman 等人提出的信任度评估模型、A. Jøsang 提出的主观逻辑信任度评估模型、George 等人提出的基于半环代数理论的信任度评估模型以及 Kamvar 等人提出的通过局部信任值迭代来计算节点的全局信任值的 EigenRep 等。

然而，分布式信任度评估模型的研究还处于起始阶段，模型复杂多样且实现存在问题，



(a) 交易历史　　(b) 当前交易

图 1.2　分布式信任度评估系统原理简图

主体之间的信任关系定义比较混乱，模型性能评价存在困难，因此在事务中还没有出现成功使用分布式信任度评估模型的案例。

1.3　信任协商

1.3.1　信任协商概述

在跨域网络协同环境中，由于交互主体间的生疏性以及共享资源的敏感性，陌生的主体之间很难建立信任关系。在信任管理系统中，资源访问请求方所提供的信任证和资源提供方所提供的策略都可能涉及敏感信息，因此在建立信任的同时，需要有效保障各方信息的隐私需求和披露自治性。为了解决上述问题，2000 年 Winsborough 等人提出了"自动信任协商"(Automated Trust Negotiation，ATN)的概念，它是"通过凭证、访问控制策略的交互披露，资源的请求方和提供方自动地建立信任关系[28]"。目前，自动信任协商作为一种新的信任管理方法，其相关研究已得到迅速发展，并成为当前的一个重要研究方向，其研究和应用在国际上备受关注。

自动信任协商主要研究跨安全域的信任建立问题，而跨域的联合协作往往属于组织频繁变化的活动，下面结合例 1.1 简要说明信任协商需要解决的主要问题，并由此引出信任协商应用系统的基本框架。

例 1.1　(医疗紧急救助）在对 Alice 实施的一次医疗急救中，急救中心 FirstAid 需要向 Alice 曾就医的医院 Hospital 请求访问其电子病历 R，然而，R 涉及 Alice 的个人隐私信息，属于敏感资源，所以，Hospital 制定了相应的保护资源 R 的访问控制策略：只有 Alice 本人及急救中心才能调阅 R。在协议消息交互过程中，各方会根据其独立的协商策略(strategy)披露相应的消息项，例如，FirstAid 只要提交当地卫生署为其签发的急救中心信任证，就能快速地访问到病历 R。

因此，建立跨安全域之间的信任通常面临着以下几个重要问题：

(1) 当隶属类似于例 1.1 中的机构 FirstAid 和 Hospital 两个独立安全域的陌生主体进行资源访问时，如何提供一种有效的方法和机制，以动态地建立两者的信任关系？

(2) 当开放网络中的协商主体在维护其自治性和隐私性时，需要什么样的访问控制策略和信任证(如例 1.1 中 Hospital 制定的访问控制策略和 FirstAid 拥有的信任证)？

(3) 对资源的访问控制结论不再是单纯的 Yes 或 No,需要根据各自的协商策略给出相应的提议,以支持进一步的协商:既要实施信息保护,又要达成联合协作。因此,如何建立协商策略机制以兼顾二者的要求?

(4) 此外,信任的建立将依赖于一套完整的协议,例如在例 1.1 中体现为机构 FirstAid 和 Hospital 的消息交互过程。

为了解决这些问题,自动信任协商研究工作需要涵盖以下 4 个方面的主要内容:体系结构及基础模型、访问控制策略及信任证、协商策略、协商协议。根据上面实例中讨论的应用需求,图 1.3 给出了一个自动信任协商应用系统的基本框架。

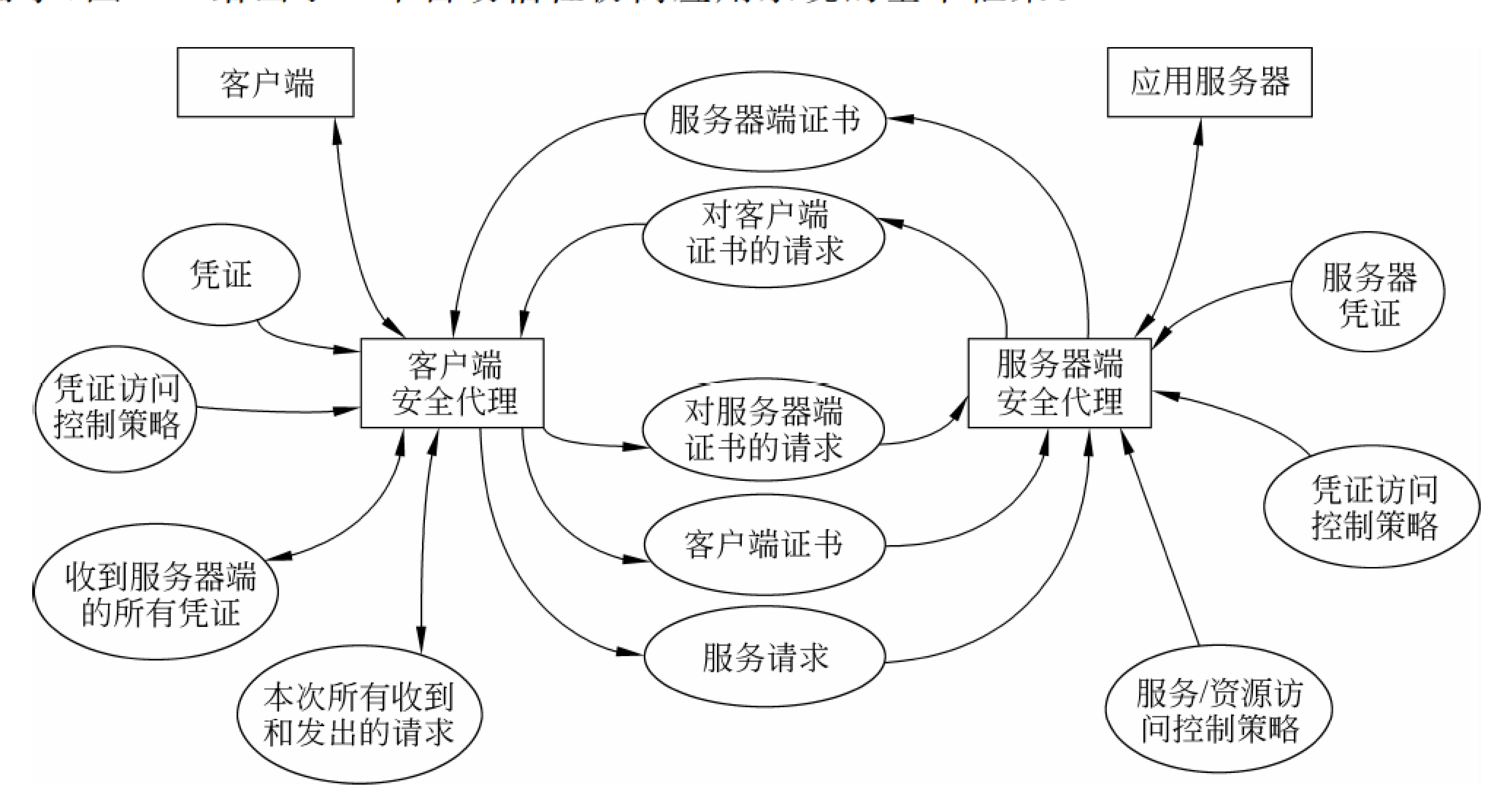


图 1.3 自动信任协商系统的基本框架

1.3.2 信任协商关键技术

从图 1.3 中可知,一个自动信任协商系统的基本要素包括信任凭证、访问控制策略、一致性校验器和信任协商策略等,这些基本要素涉及的技术也是信任协商中的关键技术。

1. 信任凭证

信任凭证是证书颁发者对证书持有者的一个带有数字签名的断言。利用现在的加密技术可以保证证书不可伪造并且能够验证。证书是利用颁发者的私钥签名并可以使用颁发者的公钥进行验证的。一个证书包含了证书持有者的一个或者多个属性,属性是由"名称/值"对构成的颁发者对持有者的一些属性描述。每一个证书也包含了证书持有者的公钥。证书持有者可以通过私钥来证明自己对证书的所有权,也可以利用自己的私钥为第三方颁发证书。

证书链的建立是通过 A 给 B 颁发证书,而 B 又以此证书为签名为 C 颁发证书的过程实现的。通过证书链的形式,总可以追溯到一个已经熟识的实体,这个实体拥有此证书链的根证书。而拥有证书链最后一个证书的主体则是需要取得信任的主体[28]。

2. 访问控制策略

在自动信任协商中,无论是服务、资源还是凭证都可以是受保护的。可以为其定义相应的访问控制策略。这一点继承了信任管理的机制,受保护的对象增多了。在自适应信任协

商中，根据保护对象的不同可以将访问控制策略分为两种：服务控制策略（Service Governing Policy，SGP）和凭证访问策略（Credential Access Policy，CAP）[28]。前者用于保护敏感的服务或资源，后者的保护对象是包含敏感信息的信任凭证。虽然这两种策略所保护的对象是不同的，但策略内容是相同的，它们都规定了对方所提交的凭证满足什么样的要求才能够访问其保护的对象。服务控制策略是服务提供者为其所提供的服务（或资源）指定相应的访问控制策略，当请求者对这些资源提出访问请求后，服务提供者会要求请求者提交相应的证书来满足其访问请求所对应的访问控制策略。凭证访问策略则是信任协商者为敏感的信任凭证所定义的访问控制策略。

3. 一致性校验器

一致性校验器是信任协商的重要组成部分，它可以判定给定的信任凭证是否能够满足针对请求资源的本地策略，从而决定是否允许对方访问资源。在一致性校验器验证证书是否满足访问控制策略时，一致性校验器首先验证信任凭证的有效性，进行匹配时过滤掉无效的证书。

传统的一致性校验器实现其基本功能。当一致性校验器收到一组信息时，信息内容主要包括信任凭证集合、访问控制策略以及对某资源或服务的请求，一致性校验器检验凭证是否有效，有效凭证集合是否满足本地的访问控制策略。根据验证的结果对请求者的请求作出响应，决定对请求者提供什么样的服务，或者是否提供服务。

自动信任协商对一致性校验器提出了更高的要求，自适应信任协商要求协商双方在一致性校验失败时，即对方提供的信任凭证不能满足本地的访问控制策略时，给对方有价值的反馈信息引导信任协商的进行，由于满足一条访问控制策略的信任凭证集合往往不止一个，所以当一方提供的凭证集合不能满足对方的访问控制策略时，并不代表此次协商没有成功地路径，这时需要给提供凭证的一方有价值的反馈信息才能引导信任协商的继续。

4. 协商策略

在信任协商框架中，协商策略引导一个信任协商的成功。协商策略控制暴露哪些证书，什么时候暴露这些证书，请求哪些证书来解锁本地的证书。并不是所有的情况下信任协商都能成功。信任协商过程中可能存在以下情况，协商一方不具备需要的证书，双方的证书的访问控制策略存在循环依赖。协商策略决定请求者什么时候放弃协商会话。协商策略必须具备以下特性：首先必须具备完整性，当一次协商存在成功的路径时，协商策略能够引导协商成功地进行，当协商不能成功时，能够终止信任协商，避免暴露不必要的信任凭证。协商策略还应该具备高效性。一个协商策略是一个函数，输入是当前的协商状态，输出是一方向另一方显示的一个信任凭证和访问控制策略的集合。

Winsborough 等人在文献[29]中介绍了两种协商策略：热心策略和吝啬策略。热心策略中，双方互相发送自己不受保护的证书，从而进一步解锁更多的证书，当客户端收到的证书无法再解锁更多的证书且无法满足服务访问控制策略时终止信任协商。证书交换的次数依赖于双方拥有的证书数量，以及双方最长的证书依赖链长度。吝啬策略中，首先披露服务访问请求以及服务访问控制策略。如果请求方存在满足服务访问控制策略的证书集，且证书集不敏感，则披露证书集。否则披露对应的访问控制策略，服务方做同样的处理，披露足够的访问控制策略直到存在不受保护的证书可以满足策略。当存在满足策略的不受保护证

书时，请求方重新发送先前的请求，披露相应的证书来解锁这些请求。

Yu 等人[29]在对上述两类策略进行研究的基础上，提出了削减(prunes)协商策略，该策略属于一种改进型的回溯策略，按深度优先方式对“安全披露序列”空间进行搜索。由于削减策略是一种暴力搜索策略，虽然完备，但搜索代价颇为昂贵。

1.3.3 信任协商方案

目前，信任协商作为一种新的信任管理方法得到了国际学术界的广泛关注，并成为当前的一个重要研究方向。BYU(Brigham Young University)的 ISRL 实验室的 Seamons 和 UIUC 大学 Winslett 等人联合承担了 ATN 的研究项目 TrustBuilder[30]，他们开展了大量的研究工作，奠定了扎实的应用基础。意大利 Milan 大学数据库与安全实验室提出了适用于 P2P 环境的信任协商框架 Trust-X[31]，研究了支持信任协商各个阶段的管理平台。此外，IBM 公司 Haifa 研究院的 TrustEstablishment 项目[32]、德国 Hannover 大学的 PeerTrust 项目[33]也在积极从事相关的研究和应用工作。

1. TrustBuilder[30,34]

杨百翰大学(BYU)的互联网安全研究室(Internet Security Research Lab，ISRL)研发出了一种支持安全代理(Security Agent，SA)之间的信任协商系统，称为 TrustBuilder 系统。每个 SA 包含访问控制策略、被保护的资源以及信任协商中间件。访问控制策略管理需保护的资源，其中被保护资源有服务、凭证和访问控制策略；信任协商中间件包括协商管理，策略独立、语言独立的协商协议以及协商策略库。其中信任协商协议定义交换消息的顺序、消息内容及信息类型，从而保证了 TrustBuilder 结构中所定义策略的互操作性；信任协商策略决定凭证是否与当前协商阶段有关以及控制交换消息的内容，如公开哪些资源，什么时候公开资源，什么时候终止协商等。TrustBuilder 系统采用绑定树策略集(Binding Tree Strategy，BTS)，从而保证最大程度上满足所需策略。在 TrustBuilder 系统中，协商双方均有一个管理协商的安全代理，用于调节访问被保护的资源。在协商过程中，安全代理通过本地策略决定下一步公开哪些策略，接收哪些新策略，以及是否终止信任协商。TrustBuilder 系统的安全代理框架如图 1.4 所示。

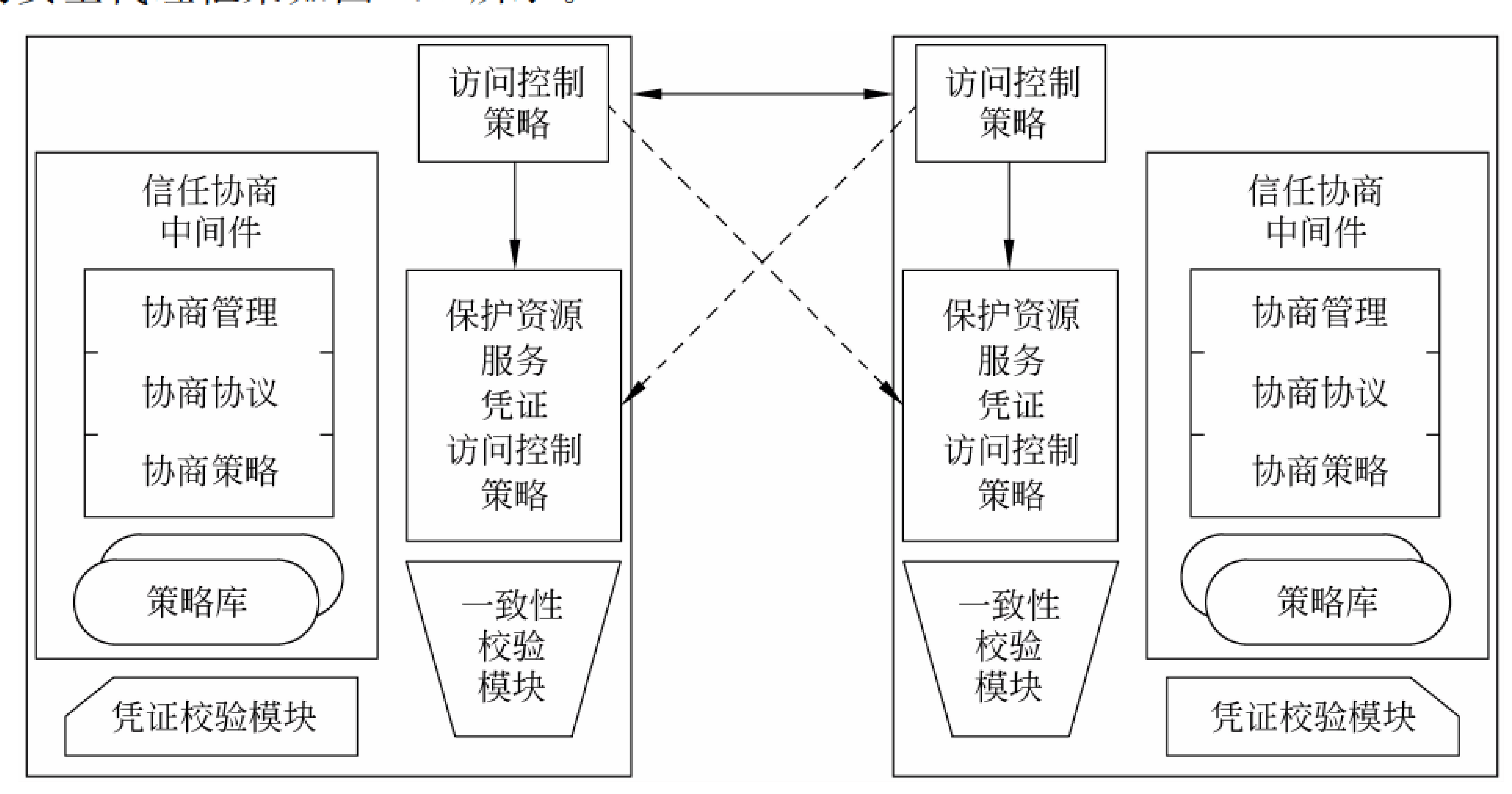


图 1.4　TrustBuilder 的安全代理框架

2. TrustEstablishment[32]

TrustEstablishment 系统是 IBM 公司的海发研究实验室(Haifa Research Lab)研究人员提出的,该系统建立的前提条件之一就是任何时候都可以公开所需要的凭证,它是根据规定约束公钥证书内容的策略在陌生人之间建立信任的系统。TrustEstablishment 系统的实体包括资源持有者、请求者和凭证发布者;TrustEstablishment 系统的数字凭证包括发布者的公钥、请求者的公钥、凭证类型和凭证版本等内容;TrustEstablishment 系统的信任策略语言(Trust Policy Language,TPL)采用 XML 语言格式,利用 XML 的灵活性对安全信息及策略进行编码,主要在分布式环境中说明和管理基于角色的访问控制。基于角色的访问控制的工作过程,系统先检测数字凭证,然后按规则将凭证持有者映射成角色,其中角色是指代表一定组织的实体,策略主要描述实体与角色的映射规则。

TrustEstablishment 系统包括 4 个主要部件:凭证库、策略引擎、智能凭证收集器以及凭证数据库。凭证库用于存放有关的数字凭证;策略引擎是 TrustEstablishment 系统的核心部分,其功能是决定一个实体能否映射成一个策略群组;智能凭证收集器可自动从凭证库中找回丢失的凭证并允许使用标准浏览器,但是浏览器只能通过一个凭证访问服务器,所以说 TrustEstablishment 系统最大特点就是支持传递策略和发现凭证链。凭证数据库用于存放与策略有关的数据,例如客户端凭证、凭证的发布者及其他数据。

由于 TrustEstablishment 系统建立的前提条件是任何时候可公开所需凭证,所以该系统的隐私保护机制不是很好,同时在 TPL 语言和系统结构设计上没有敏感性策略的概念。

3. Trust-X[31]

Trust-X 是 Elisa Betino 等人提出的一种适用于 P2P 环境的信任协商模式。上面提到的信任协商系统都考虑到策略和信任凭证说明、协商策略的选择,但是缺少一个好的协商管理平台对信任协商的各个阶段进行说明。Trust-X 系统正好弥补了这一缺点。它对信任协商过程的所有阶段进行考虑并且提出了一个信任协商的综合解决方法。

Trust-X 协商框架如图 1.5 所示,该结构是均衡对等的,它支持的主要功能有:策略互换、测定一个策略是否可以得到满足、支持凭证和信任票互换及序列缓存,结构中的一个部件支持一种功能,如序列预置模块用于存储最近成功协商过的信任队列,当下次实体请求某资源进行协商时,若序列预置模块中存在相似的信任协商序列,则从中选择序列进行协商,从而加快协商速度;协商树管理器用于记录信任协商过程中每一个协商状态,通过一定算法,在协商树中可以找出一种或多种信任协商序列;一致性校验器包括一个凭证验证模块,对接收的凭证作有效性检查,其目的是在必要的情况下验证文档签名、信任凭证的撤回以及发现凭证链。

该系统还具有许多独特的功能,比如支持信任票、支持敏感性策略保护、支持预置策略以及策略选择等功能。信任票也是一种凭证,当协商双方成功协商后,该系统会给双方发送一个信任票,当以后双方再次为某资源协商时,系统会先检测双方是否存在信任票,若存在信任票的情况,说明双方已经建立过信任,从而不必进行信任协商就可获取资源,这就加快了协商进程。但是信任票的有效性是有时间限制的,这个时间限制主要取决于双方的属性特征,只有当信任票处于有效期内,信任票才有效。Trust-X 系统还引入了预置策略的概念,它是指公开策略 B 之前必须先公开策略 A,此时称策略 A 是策略 B 的预置策略,从而保

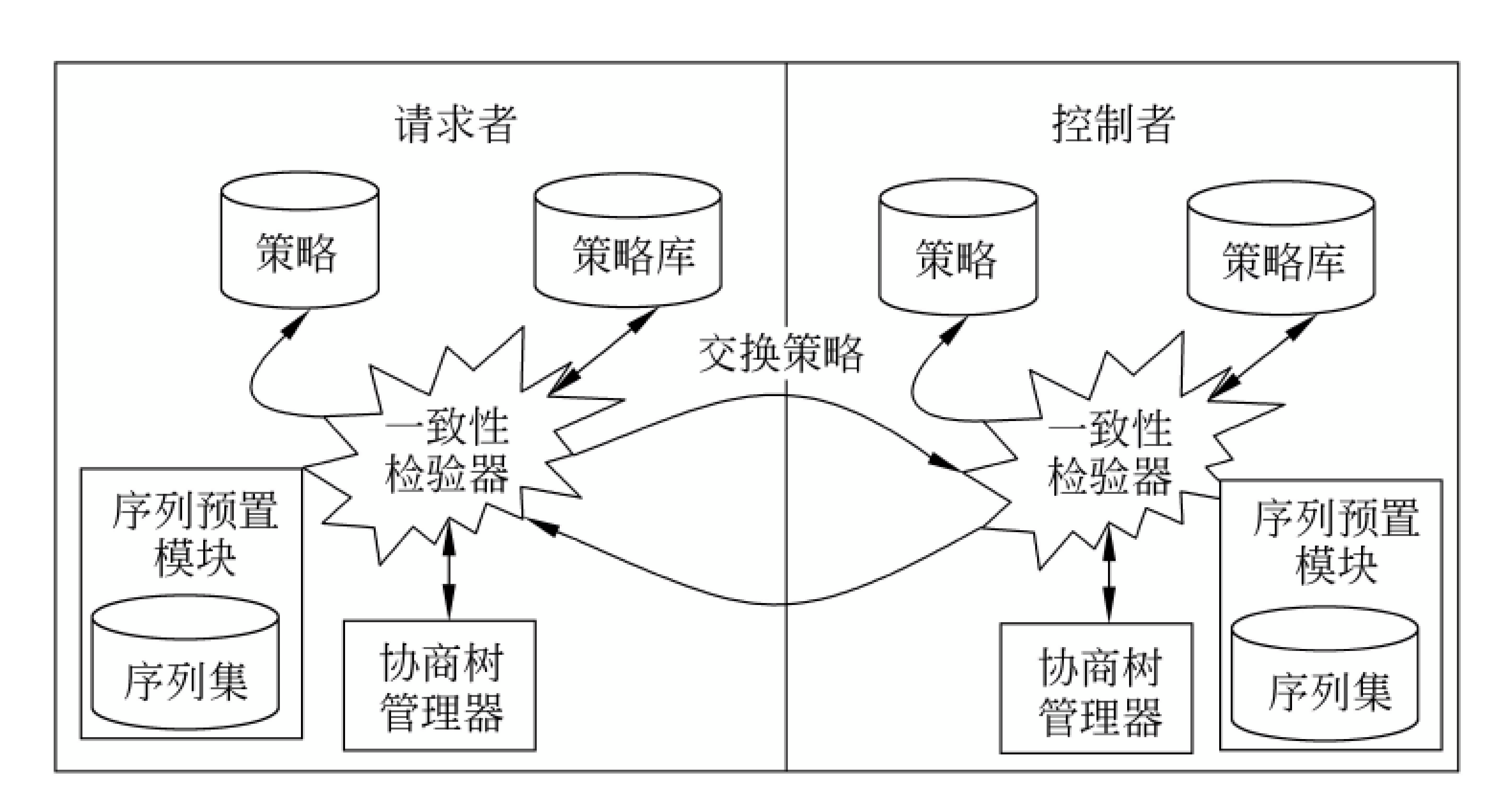


图 1.5 Trust-X 信任协商框架

护敏感性的公开策略。

Trust-X 的语言称为 X-TNL 语言，它与 TPL 语言一样都是基于 XML 编码的，用来描述 Trust-X 数字凭证和公开策略，也就是说在 X-TNL 中，一个信任凭证的类型是 DTD 模式，但 X-TNL 语言支持上面所提到的信任票。Trust-X 系统支持阶段性协商，协商方持有信任票协商可直接获得资源。

模块中存有相似协商序列的协商也可快速获取资源，但是对于敏感度高的资源则必须进入策略评估阶段，从敏感性低的策略逐步向敏感性高的策略协商。

1.4 本章小结

本章对信任管理的思想方法和技术要领进行了简明扼要的阐述，首先介绍了信任和信任管理的基本概念，讨论了信任管理的产生背景、基本内涵、主要内容以及国内外研究现状。目前信任度评估和信任协商是信任管理领域的两个重要研究方向，本章概要地讨论了信任度评估涉及的主要内容，介绍了信任协商的基本概念、关键技术和解决方案，以为本书后续章节作铺垫。

参 考 文 献

[1] D. Gambetta. Can we trust? Trust: Making and Breaking Cooperative Relations, 1988: 213-238.

[2] K Konrad, G Fuchs, J Bathel. Trust and electronic commerce-more than a technical problem. In: The 18th symposium on reliable Distributed systems, 1999.

[3] A Kini, J Choobineh. Trust in electronic commerce: definition and theoretical considerations. In: 31st Hawaii International conference on System Sciences, 1998.

[4] Grandison, T, M. Sloman. A Survey of Trust in Internet Applications. IEEE Communications Surveys and Tutorials, 2000. 4(4).

[5] Tyrone W, A Grandison. Trust management for Internetapplications: Ph. D. Thesis[D]. University of London, 2003.

[6] L Mui. Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks. PhD Dissertation, Massachusetts Institute of Technoligy, 2003.

[7] Y wang, J Vassileva. Trust and Reputation Model in Agent-to-Agent Network. In: Proceeding of the 3rd IEEE International Conference on Peer-to-peer Computing, 2003. Washington D C: IEEE Computer Society, 2003: 150-157.

[8] EJ Chang, FK Hussain, TS Dillon. Fuzzy nature of trust and dynamic trust modeling in service oriented environments. In: Proceedings of the 2005 workshop on Secure Web Services. Farifax, USA: ACM Press, 2005: 75-83.

[9] McKnight D H, Chervany N L. Conceptualizing Trust: a Typology and E-commerce Customer Relationships Model [A]. In: Proceedings of the 34th Annual Hawaii International Conference on System Sciences [C]. Washington, DC, USA: IEEE Computer Society, 2001:1-10.

[10] Blaze, M., Feigenbaum, J., Lacy, J. Decentralized trust management. In: Dale, J., Dinolt, G., eds. Proceedings of the 17th Symposium on Security and Privacy. Oakland, CA: IEEE Computer Society Press, 1996. 164-173.

[11] Blaze, M., Feigenbaum, J., Keromytis, A. D. KeyNote: trust management for public-key infrastructures. Cambridge 1998 Security Protocols International Workshop. Berlin: Springer-Verglag, 1999: 59-63.

[12] Abdul-Rahman, A., Hailes, S. A distributed trust model. In: Proceedings of the 1997 New Security Paradigms Workshop. Cumbria, UK: ACM Press, 1998: 48-60.

[13] Gambetta, D. Can we trust trust? In: Gambetta, D., ed. Trust: Making and Breaking Cooperative Relations. Basil Blackwell: Oxford Press, 1990: 213-237.

[14] Povey, D. Developing electronic trust policies using a risk management model. In: Proceedings of the 1999 CQRE Congress. 1999: 1-16.

[15] J. P Anderson. Computer Security Threat Monitoring and Surveillance. Technical report, James PAnderson Co., Fort Washington, Pennsylvania, April 1980.

[16] Stephen Paul Marsh. Formalizing Trust as a Computational Concept[dissertation], University of Stirling, Scotland, U. K., 1994.

[17] Abdul-Rahmana, Hailes S. Supporting trust in virtual communities [C]. In: Proc of the Hawaii International Conference on System Sciences. LosAlamitos: IEEE ComputerSociety, 2000.

[18] Stephen Paul Marsh. Formalising Trust as a Computational Concept [M]. Dept. of Computing Science and Mathematics, University of Stirling, 1995.

[19] 常俊胜，王怀民，尹刚. DyTrust:一种P2P系统中基于时间的动态信任模型[J]. 计算机学报，2007，29(8): 1301-1307.

[20] Jøsang A, Ismail R. The beta reputation system [J]. In: Proeeedings of the l5th Bled Electronic Commerce Conference, June 2002.

[21] WANG Yao, VASSILEVA J. Bayesian network-based trust model[C]. In: Proc of IEEE/WIC International Conference on Web Intelligence. Halifax: IEEE Press, 2003: 372-378.

[22] 陈建刚，王汝传，王海艳. 网络资源访问的一种主观信任机制[J]. 电子学报，2007，34(5):817-821.

[23] Song Shan-shan, Wang K, Zhou Run-fang. Trusted P2P transactions with fuzzy reputation aggregation [J]. IEEE Internet Computing, 2007, 9(6):24-34.

[24] Ramchurn S D, Sierra C, Godo L. Devising a trust model for multi-agent interactions using confidence and reputation [J]. International Journal of Applied Artificial Intelligence, 2003, 18(10):833-852.

[25] Dempster A P. A Generalization of Bayesian Inference [J]. Journal of the Royal Statistical Society, 1968, Series B 30:205-245.

[26] 杨风暴，王肖霞. D-S证据理论的冲突证据合成方法[M]. 北京：国防工业出版社，2010: 2-3.

[27] Winsborough WH, Seamons KE, Jones VE. Automated trust negotiation. In: DARPA Information Survivability Conf. And Exposition. New York: IEEE Press, 2000. 88-102.

[28] Winsborough WH, Seamons KE, Jones VE. Automated trust negotiation. In: DARPA Information Survivability Conf. and Exposition. New York: IEEE Press, 2000: 88-102.

[29] Yu T, Ma X, Winslett M. PRUNES: An efficient and complete strategy for trust negotiation over the Internet. In: Proc. of the 7th ACM Conf. on Computer and communications Security. New York: ACM Press, 2000: 210-219. http://www4.ncsu.edu:8030/~tyu/pubs/ccs2000.pdf.

[30] Smith B, Seamons KE, Jones MD. Responding to policies at runtime in TrustBuilder. In: Proc. of the 5th Int'l Workshop on Policies for Distributed Systems and Networks. Washington: IEEE Computer Society Press, 2004: 149-158.

[31] Bertino E, Ferrari E, Squicciarini A. Trust-X: A Peer-to-Peer framework for trust establishment. IEEE Transaction on Knoledge and Data Engineering, 2004, 16(7): 827-842.

[32] Herzberg A, Mass Y, Michaeli J, Ravid Y, Naor D. Access control meets public key infrastructure, or: Assigning roles to strangers. In: Proc. of the 2000 IEEE Symp. on Security and Privacy. Washington: IEEE Computer Society Press, 2000: 2-14.

[33] Basney J, Nejdl W, Olmedilla D, Welch V, Winslett M. Negotiating trust on the grid. In: Proc. of the 2nd Workshop on Semantics in P2P and Grid Computing at the 13th Int'l World Wide Web Conf. 2004.

[34] M. Winslett, Yu T, K. E. Seamons, et al. Negotiating trust on the web [J]. IEEE Internet Computing, 2002, 6(6): 30-37.

[35] 陈文亮. 一种基于证据的信任度评估模型研究[D]. 北京信息科技大学, 2011.

[36] Jøsang. A. A model for trust in security systems. In: Proceedings of the 2nd Nordic Workshop on Secure Computer Systems, 1997.

第 2 章　基于多维证据的信任度评估模型

在第 1 章中介绍了信任度评估的主要思想方法，信任度评估是基于行为的信任管理模型关注的核心内容。近年来，众多学者在该领域做了许多有益的研究，也提出了不少信任度评估模型，但总体来看，这些模型一般都只是把网络主体参与特定交易的交易反馈信息作为评估的基础数据来源，而没有考虑主体通过技术手段实施的网络操作行为，使得信任度评估所依据的证据源不够完备。

本章提出一种基于多维证据的信任度评估模型，该模型将主体的网络操作行为层面的信息引入传统仅考虑交易反馈信息的信任度评估模型，基于交易反馈和网络操作行为两个层面的多维证据源进行信任计算，扩展了证据源，突破了只依据单一种类证据源进行信任评估而引起的缺陷；另外，我们应用改进的 D-S 证据理论来合成多维证据，能够很好地解决证据不确定性的问题。

2.1　多维证据

第 1 章已经介绍了信任度评估的证据，现有信任度评估模型中最常出现的信任因子主要有交易额、交易量、交易时间、交易结果和交易评价等。而这些证据大多可归为基于业务反馈的证据。这些证据源有明显的局限性，主要表现在以下两点：

(1) 忽视了与交易业务相关的网络操作活动对交易的影响。比如，网络主体在进行某项交易前，通过某些黑客手段获得了交易优势(比如在电子商务系统中篡改自己或他人的信誉值等)，这种行为本身体现了该主体的不可信，但是这些信任并不能在业务反馈证据中得以体现。

(2) 只考察主体在某特定类型交易中的表现。如果一个人在某一种活动中是诚信的，那么我们会倾向于认为他在其他活动中也是诚信的。因此，考察某网络主体在 A 类交易中的信任度，对于评价该主体在 B 类交易中的信任度是有一定意义的。

鉴于以上局限性，本节从评价某一个网络主体可信度的视角出发，提出了多维证据的概念，并基于多维证据提出了一种基于多维证据的信任度评估模型。多维证据中所谓多维是指不同类型的证据，本章涉及的多维证据主要包括电子商务类业务反馈证据、网络社区类业务反馈证据和网络操作行为证据。

2.1.1　电子商务类业务反馈证据

电子商务类业务反馈证据是指网络主体参与在线买卖交易的相关证据，当前已经有很多信任度评估模型基于这一类证据，因此对于此类证据的研究已经比较充分。在线交易是网络主体参与网络活动的一项重要内容，网络主体在线交易中表现出来的诚信状态对其信任度产生重要影响。

本章将一条电子商务类业务反馈证据的数据结构定义为：证据（证据类别，主体身份，交易时间，交易价值，交易结果，交易评价），简写为 Evi（Cla，Ide，T，Val，Res，Asse）。其中，Evi 表示证据名称（Evidence），只是一个符号，说明此记录是一条证据；Cla 表示此证据的类别（Classification），用于标识当前证据属于多维证据中的哪一类；Ide 表示主体身份（Identity），用于标识当前主体在当前交易中的角色，分为卖者和买者两种；T 表示交易时间（Time），用于标识当前交易发生的时间；Val 表示交易价值（Value），用于标识当前交易标的物的价值；Res 表示交易结果（Result），用于标识当前交易的结果，分为成功、失败；Asse 表示交易评价（Assessment），用于标识当前主体在本次交易中获得的评价，分为正面评价、中性评价和反面评价。

2.1.2 网络社区类业务反馈证据

网络社区类业务反馈证据是指网络主体在其加入的网络社区进行的相关活动的数据。加入网络社区并参与发帖活动是网络主体参与网络活动的又一重要内容，网络主体在网络社区中的表现对其信任度也产生重要影响。

本章将一条网络社区类业务反馈证据的数据结构定义为：证据（证据类别，事件，时间，事件判别），简写为 Evi（Cla，Eve，T，Disti）。其中，Cla 表示此证据的类别（Classification），用于标识当前证据属于多维证据中的哪一类；Eve 表示事件（Event），包括主页被浏览、原发帖被浏览、被删帖和被禁止发帖；T 表示时间（Time），记录此证据产生的时间；Disti 表示对事件的判别（Distinguish），用于对事件进行判断定性。其中，对于主页或帖子被浏览事件，Disti 判别累计次数是否达到某阈值，如果达到则产生一条证据记录；对于被删帖事件，Disti 取值包括楼主和回复；对于被禁止发帖事件，Disti 的取值包括暂时禁止和封号。

2.1.3 网络操作行为证据

网络操作行为是指在网络技术层面上表现的行为，包括正常行为和入侵行为；其中，入侵行为是指诸如非法访问、口令猜测、DOS 攻击和木马攻击等危害网络安全的行为，这些行为与交易相关并可能破坏交易公平性或者危害网络安全。网络主体实施危害性网络操作行为，将对其信任产生较为严重的破坏性影响。

本章提出的信任度评估模型也关注网络主体过去的行为，但不像防火墙和入侵检测系统那样对网络进行实时保护，而是基于安全审计技术来分析和发现网络操作行为证据。安全审计系统可以对网络中各种设备和系统进行集中的审计，发现入侵行为及安全隐患。本章利用安全审计结果中关于网络主体的入侵行为，并获取该主体的网络操作行为证据。

本章将一条网络操作行为证据的数据结构定义为：证据（证据类别，时间，攻击级别），简写为 Evi（Cla，T，Lev）。其中，Cla 表示此证据的类别（Classification），用于标识当前证据属于多维证据中的哪一类；T 表示时间（Time），记录此证据指向的网络操作行为实施的时间；Lev 表示攻击级别，本章将攻击级别分为 5 个等级[1]，即 Lev＝1 表示信息泄露类攻击，Lev＝2 表示拒绝服务类攻击，Lev＝3 表示数据破坏和欺骗类攻击，Lev＝4 表示入侵控制类攻击，Lev＝5 表示对抗性攻击。

2.2　D-S 证据理论及合成规则改进

2.2.1　D-S 证据理论的基本原理

D-S 证据理论是对概率论的扩展，概率论可以视为 D-S 证据理论的一个特例。D-S 证据理论将命题的推理转化为集合的推理，当一个证据 E 不能完全支持或者拒绝一个命题 A 时，将 E 所包含的信息分配给包含 A 的一个集合，从而使得 E 所包含的信息不至于丢失，同时也描述了 E 对支持或拒绝 A 的不确定性。当一个证据对一组命题支持或者拒绝的先验概率已知时，D-S 证据理论与概率论将产生相同的推理结果。

D-S 证据理论引入识别框架、基本概率分配函数、信任函数和似然函数等概念来描述不确定性，通过对证据的合并和对信任函数的更新来实现集合的推理。

1. 识别框架

定义 2.1　识别框架

对于一个需要判决的问题，其答案的完备集合用 Θ 表示，Θ 中的元素可以是数值，也可以是非数值，Θ 中所有元素都是两两互斥的，且任意时刻问题的答案都只能取 Θ 中的某一个元素，则称 Θ 为该问题的识别框架。识别框架 Θ 可表示为

$$\Theta = \{\theta_1, \theta_2, \cdots, \theta_i, \cdots, \theta_n\} \tag{2.1}$$

其中，θ_i 是识别框架 Θ 的一个元素，n 是 Θ 中元素的个数，$i \in [1, n]$。

由识别框架 Θ 所有子集组成的集合称为 Θ 的幂集，记作 2^Θ，可表示为

$$2^\Theta = \{\varnothing, \{\theta_1\}, \{\theta_2\}, \cdots, \{\theta_1, \theta_2\}, \{\theta_1, \theta_3\} \cdots, \Theta\} \tag{2.2}$$

例如，判断一个日期所处的季度，那么该问题的识别框架为：Θ＝{第一季度，第二季度，第三季度，第四季度}，Θ 的子集{第一季度，第二季度}表示这一天是第一季度或者第二季度的某一天。

2. 基本信任分配函数

定义 2.2　基本信任分配函数

如果函数 m 满足下列条件的映射：

$$\begin{cases} m: 2^\Theta \to [0,1] \\ \sum\limits_{A \subset \Theta} m(A) = 1 \\ m(\varnothing) = 0 \end{cases} \tag{2.3}$$

则称 m 是 2^Θ 上的基本信任分配函数，其中 $m(A)$ 称为 A 的基本信任分配函数，它表示证据对 A 的精确信任程度。

式(2.3)中，$\sum\limits_{A \subset \Theta} m(A) = 1$ 表示虽然可以给每一个命题集合分配任意信任值，但是所有命题集合所获得的信任值之和必须等于 1；$m(\varnothing) = 0$ 表示对空集不分配信任值，即对空集不产生支持或拒绝的值。

如果 $m(A) > 0$，则称 A 为焦元，焦元中包含识别框架 Θ 元素的个数称为焦元的基。基为 1，则称 A 为单元焦元；同样，基为 i，则称 A 为 i 元焦元。如果 A 是单元焦元，则 $m(A)$ 表

示当前证据对 A 的精确信任度；如果 A 是多元焦元且 $A \neq \Theta$，则 $m(A)$ 依然是当前证据对 A 的精确信任度，不同的是这个信任度无法精确地分配给 A 中的元素。如果 $A=\Theta$，那么 $m(A)$ 表示当前证据对 Θ 的子集不产生信任值，即不知道该把信任值分配给谁。

在实际应用过程中，基本信任分配函数是需要构造的，构造基本信任分配函数时通常根据过去获得的数据或者专家们的经验作为基础。

3. 信任函数

定义 2.3 信任函数

对于任意一个命题或者命题集合，其信任函数 Bel(A)定义为 A 的全部子集对应的基本信任分配函数之和，即

$$\begin{cases} \mathrm{Bel}: 2^{\Theta} \to [0,1] \\ \mathrm{Bel}(A) = \sum\limits_{B \subseteq A} m(B) \\ A \subseteq \Theta \end{cases} \tag{2.4}$$

Bel 函数又被称为下限函数，表示当前证据对于 A 的全部信任值。由基本信任分配函数容易得到 $\mathrm{Bel}(\varnothing)=m(\varnothing)=0$。如果 $m(A)>0$，则称 A 为信任函数 Bel 的焦元，所有焦元并称为 Bel 的核。

4. 似然函数

定义 2.4 似然函数

如果对识别框架 Θ 的任一子集 A，有

$$\begin{cases} \mathrm{Pl}: 2^{\Theta} \to [0,1] \\ \mathrm{Pl}(A) = 1 - \mathrm{Bel}(-A) \\ A \subseteq \Theta \end{cases} \tag{2.5}$$

则称 Pl(A)为 A 的似然函数。似然函数也被称为下限函数，表示对 A 非假的精确信任度，即表示对 A 可能成立的不确定程度。

根据信任函数和似然函数的定义，不难证明信任函数和似然函数有如下关系：

$$\mathrm{Pl}(A) \geqslant \mathrm{Bel}(A), \quad A \subseteq \Theta$$

A 的不确定性由 $u(A)=\mathrm{Pl}(A)-\mathrm{Bel}(A)$ 表示，A 的不确定性也可以用图 2.1 来表示。



图 2.1　证据对 A 的支持区间图

5. Dempster-Shafer 证据合成规则

由上面介绍的相关知识可知，对于某一命题或者命题集合成立的表述需要用信任函数和似然函数来度量，而信任函数和似然函数又是基于基本信任分配函数来定义的，所以基本信任分配函数是基础。在实际应用过程中，通常会有多个证据同时对某一命题或者命题集合产生作用，而不同的证据可能会产生不同的基本信任分配函数，所以为了计算信任函数和

似然函数，必须先将所有不同的基本信任分配函数合并成一个概率分配函数。因此，Dempster 提出了一种合成不同基本信任分配函数的方法[2,3]，通常称为 Dempster-Shafer 证据合成规则，简称为 D-S 合成规则。

1）两个证据的合成

假设 m_1、m_2 分别是识别框架 Θ 下两个证据 E_1、E_2 的基本信任分配函数，焦元分别为 A_{1i}、A_{2j}，则 D-S 合成规则为

$$m(A) = \frac{\sum\limits_{A_{1i} \cap A_{2j} = A} m_1(A_{1i}) m_2(A_{2j})}{1 - K}, \quad 且\ K < 1 \tag{2.6}$$

式中，

$$K = \sum_{A_{1i} \cap A_{2j} = \varnothing} m_1(A_{1i}) m_2(A_{2j}) \tag{2.7}$$

由式(2.6)可知，D-S 合成规则实质上是对 m_1、m_2 做正交运算，通常记为 $m = m_1 \oplus m_2$。如果 $K<1$ 不成立，那么 m_1 和 m_2 就无法用 D-S 合成规则来合成，即 $m = m_1 \oplus m_2$ 不存在。

D-S 合成规则的几何意义可以用图 2.2 来说明。图 2.2(a)和图 2.2(b)分别表示证据 E_1、E_2 的基本信任分配，图 2.2(c)表示 E_1、E_2 的基本信任分配函数的合成结果。图 2.2(c)中阴影部分表示对于该焦元 m_1、m_2 都产生了支持度，该焦元在合成后的信任分配函数中也可以获得信任分配；空白部分表示 m_1、m_2 没有同时对相应焦元产生支持度，所以这些焦元在合成后的信任分配函数中将不能获得信任分配。

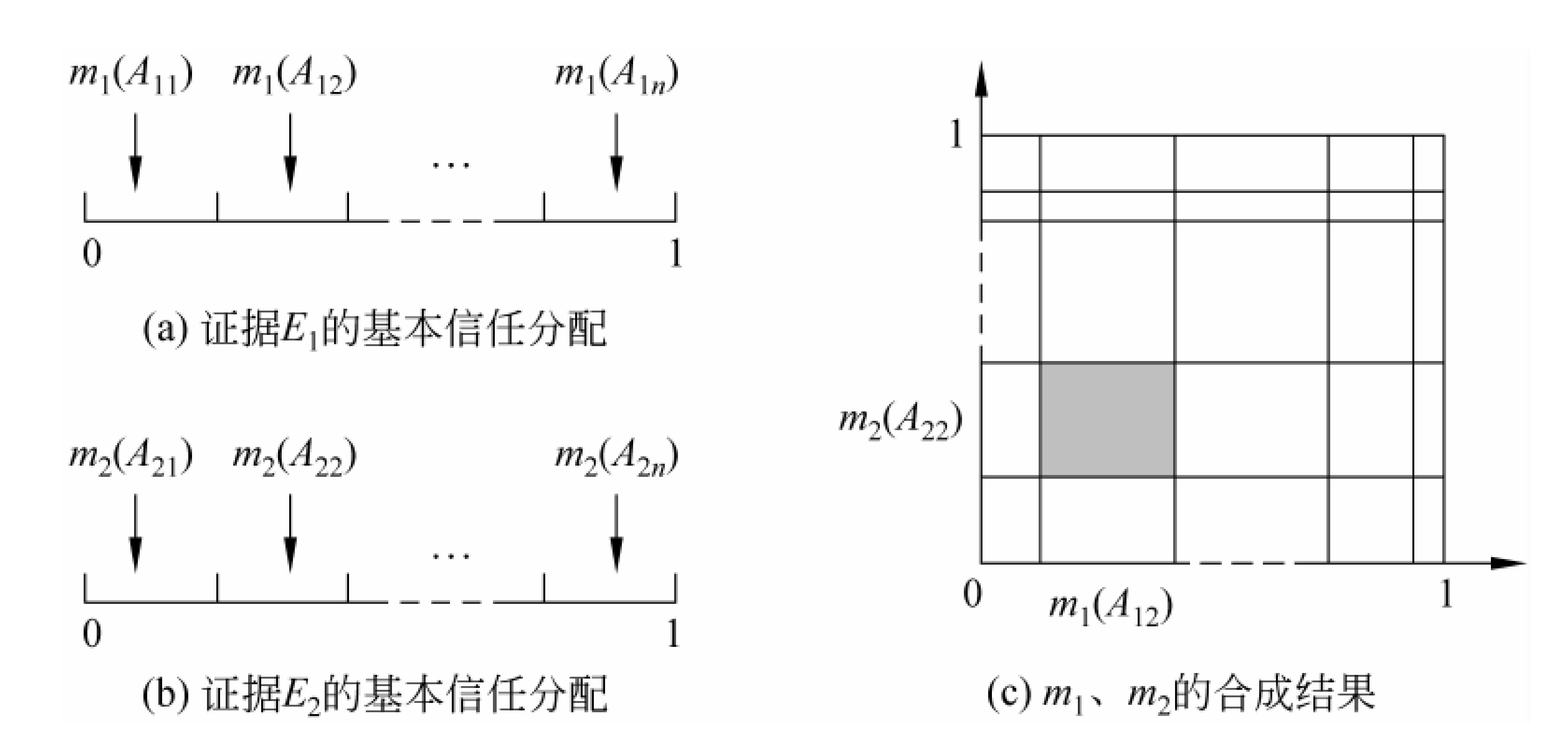


图 2.2　两个证据的基本信任分配函数合成的几何意义

2）多个证据的合成

对于 E_1、E_2、…、E_n 在识别框架 Θ 下的基本信任分配函数 m_1、m_2、…、m_n 进行合成，可以表示为 $m = m_1 \oplus m_2 \oplus \cdots \oplus m_n$，合成规则如下：

$$m(A) = \frac{\sum\limits_{\cap A_{nj} = A} \prod\limits_{\substack{1 \leqslant i \leqslant n \\ 1 \leqslant j \leqslant q}} m_i(A_{ij})}{1 - K} \tag{2.8}$$

其中，$A \neq \varnothing$，n 是证据的个数，q 是焦元的个数，$K<1$，且 K 的计算方法如下：

$$K = \sum_{\cap A_{nj} = \varnothing} \prod_{\substack{1 \leqslant i \leqslant n \\ 1 \leqslant j \leqslant q}} m_i(A_{ij}) \tag{2.9}$$

2.2.2 D-S 合成规则改进

对于 D-S 证据理论的改进主要着力于对于证据合成规则的改进，因为在实际应用过程中，D-S 合成规则会出现不能使用或者得出不符合人类推理习惯的结论。文献[4]分析总结了 D-S 合成规则可能产生的六大悖论，即全冲突悖论、0 信任悖论、1 信任悖论、证据失效悖论、信任偏移悖论和焦元基模糊悖论。D-S 合成规则产生的悖论主要是由于冲突证据的存在导致的，目前已有很多关于冲突证据合成的研究成果。例如：

(1) Smets 合成规则。在假设证据完全可靠的前提下，提出了高冲突证据的合成规则，被称为可传递置信模型[5]。Smets 认为，导致冲突证据合成悖论的主要原因是识别框架有穷且完备的假设不合理，因为人的认识是有局限性的，必然存在人们无法判断其真假的位置命题。Smets 将冲突的原因归结于未知命题的存在，并将冲突分配给这些命题，基于这种思想提出了可传递置信模型。但是这个合成规则不太符合人类的推理思维。

(2) Yager 合成规则。该规则在假设证据不完全可靠的前提下，取消了正则化过程，提出了一种证据合成规则，其主要思想是把冲突部分分配给识别框架 Θ[6]。Yager 合成规则在合成低冲突的证据时可以获得良好的结果，但是合成高冲突的证据时却不能获得满意的结果。

(3) Lefevre 合成规则。Lefevre 认为在合成冲突证据时，冲突部分应该尽可能地分配给涉及冲突的焦元[7]。基于这一思想，Lefevre 为所有涉及冲突的焦元定义了加权因子，并把冲突部分按照加权因子成比例地分配给各个涉及冲突的焦元。事实上，Lefevre 的合成规则只考虑了一种冲突，即当存在某焦元在不同证据的基本信任分配函数中获得了 0 和非 0 的信任分配。然而，这种冲突并不能描述全部的冲突，因为某焦元在不同的两个证据中分别获得 0.9 和 0.1 的信任分配也是一种高冲突，而这种冲突不在 Lefevre 合成规则的考虑范围之内。但是把冲突部分按权重分配给冲突焦元的思想很值得借鉴。

(4) Murphy 合成规则。Murphy 认为，既然 D-S 合成规则产生问题的原因是证据间的冲突，那么如果能够通过某种方法将证据间的冲突降低，则 D-S 合成规则就仍然是可行的。基于这一思路，Murphy 提出了一种改进的合成规则[8]，Murphy 的合成规则可以处理高冲突的证据合成问题，但是简单平均的方法忽视了证据之间冲突的具体情况，在实际运用中可能会产生偏离实际的结果。比如，某些偏差很大的不准确证据可能会对合成结果产生较大的搅动，使得合成结果偏离实际情况。

得益于 Murphy 的思想，同时考虑避免简单平均法带来的弊端，我们提出一种基于冲突强度 G 和有效冲突 G_h 的新合成规则，简称为 $G\text{-}G_h$ 合成规则。该合成规则的步骤为：

(1) 基于冲突强度 G，将所有证据的信任分配值进行加权平均。

(2) 运用有效冲突合成法对 n 个 $m(A_i^e)$ 进行合成 $n-1$ 次。

以下具体阐述 $G\text{-}G_h$ 合成规则。

假设 m_1、m_2、…、m_n 分别是识别框架 Θ 下证据 E_1、E_2、…、E_n 的基本信任分配函数，A_1、A_2、…、A_l 是所有证据涉及的全部焦元。

1. 冲突强度 G

文献[4]给出了冲突强度 G 的定义，即：

假设 m_1、m_2 分别是识别框架 Θ 下两个证据 E_1、E_2 的基本信任分配函数，焦元分别为 A_{1i}、A_{2j}，则有

$$\begin{cases} G(E_1,E_2)=\dfrac{C(E_1,E_2)}{H(E_1,E_2)+C(E_1,E_2)} \\ C(E_1,E_2)=\sum\limits_{A_{1i}\cap A_{2j}=\varnothing} m(A_{1i})m(A_{2j}) \\ H(E_1,E_2)=\sum\limits_{A_{1i}=A_{2j}} m(A_{1i})m(A_{2j}) \end{cases} \tag{2.10}$$

2. 加权平均信任分配值

对于任意两个证据 E_i、E_j，定义 $b_{ij}=1-G_{ij}=b_{ji}$ 为证据 E_i、E_j 的相似度，且 $b_{ii}=b_{jj}=1$，由此可以获得所有 n 个证据的相似矩阵：

$$\boldsymbol{S}=\begin{pmatrix} b_{11} & \cdots & b_{1j} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{i1} & \cdots & b_{ij} & \cdots & b_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nj} & \cdots & b_{nn} \end{pmatrix}=\begin{pmatrix} 1 & \cdots & b_{1j} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{i1} & \cdots & 1 & \cdots & b_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nj} & \cdots & 1 \end{pmatrix} \tag{2.11}$$

证据间冲突强度越小，说明证据相互支持的力度越大，因此矩阵 $\boldsymbol{S}$ 每行元素相加可以得到其他所有证据对 E_i 的支持力度，即

$$\mathrm{Sup}(m_i)=\sum_{j=1}^{n} b_{ij},\quad (i,j=1,2,\cdots,n) \tag{2.12}$$

将支持度归一化可得加权平均的权重，即

$$\mathrm{Weit}(m_i)=\mathrm{Sup}(m_i)\Big/\sum_{i=1}^{n}\mathrm{Sup}(m_i),\quad (i,j=1,2,\cdots,n) \tag{2.13}$$

此时，可以对 n 组证据的基本信任分配值进行加权平均，即

$$m(A_l^e)=\sum_{i=1}^{n}\mathrm{Weit}(m_i)m_i(A_l) \tag{2.14}$$

3. 运用有效冲突合成法做最终合成

经过以上步骤，可以获得一个加权平均后的基本信任分配函数，即

$$m^e(A)=m^e(\{A_1\},\{A_2\},\cdots,\{A_l\})=(m^e(A_1),m^e(A_2),\cdots,m^e(A_n)) \tag{2.15}$$

Murphy 使用 D-S 合成法（即式(3.6)和式(3.7)）对平均后的证据进行合成，该方法利用归一化因子 $1/(1-k)$ 将冲突部分完全分配给非 Θ 焦元。这一思想并非完全合理，因为冲突部分 k 并非为完全有效信息，如果将无效信息分配给非 Θ 焦元，那么合成结果实际上已经偏离了正确结果。

针对以上问题，本章将冲突部分划分为有效冲突和无效冲突两部分，有效冲突部分是指对于命题判别过程有效的部分，无效冲突部分是指对命题判别过程无效的部分。基于文献[4]给出的证据一致量和冲突量的定义，本章给出冲突有效部分的定义。

定义 2.5 证据一致量、冲突量和冲突有效部分

对于识别框架 Θ 下两个证据 E_1、E_2，基本信任分配函数分别是 m_1、m_2，焦元分别为 A_{1i}、A_{2j}，A_1、A_2、…、A_l 是所有证据涉及的全部焦元，则有：

(1) E_1、E_2 的证据一致量

$$H(E_1,E_2)=\sum_{A_{1i}=A_{2j}} m(A_{1i})m(A_{2j}) \tag{2.16}$$

(2) E_1、E_2 的证据冲突量

$$C(E_1,E_2)=\sum_{A_{1i}\cap A_{2j}=\varnothing} m(A_{1i})m(A_{2j}) \tag{2.17}$$

(3) E_1、E_2 的冲突有效部分

$$G_h(E_1,E_2)=\frac{H(E_1,E_2)}{H(E_1,E_2)+C(E_1,E_2)} \tag{2.18}$$

基于上述定义，本章给出有效冲突合成法。假设 m_1、m_2 分别是识别框架 Θ 下两个证据 E_1、E_2 的基本信任分配函数，焦元分别为 A_{1i}、A_{2j}，A_1、A_2、…、A_l 是所有证据涉及的全部焦元，则有

$$m(A_i)=\begin{cases}\sum\limits_{A_{1i}\cap A_{2j}=A_l} m_1(A_{1i})m_2(A_{2j})+\Delta\varphi\times G_h\times K, A\neq\Theta\\ \sum\limits_{A_{1i}\cap A_{2j}=A_l} m_1(A_{1i})m_2(A_{2j})+(1-G_h)\times K, A=\Theta\end{cases} \tag{2.19}$$

式中，

$$\Delta\varphi=\begin{cases}\dfrac{m_1(A_l)+m_2(A_l)}{\sum m_1(A_{1i})m_2(A_{2j})}, & A_l,A_{1i},A_{2j}\text{ 为非 }\Theta\text{ 焦元}\\ 0, & A_l=\Theta\end{cases} \tag{2.20}$$

(4) 对 n 个 $m^e(A)$ 合成 $n-1$ 次的新算法

$G\text{-}G_h$ 合成规则保留了 D-S 合成的交换律和结合律，因此本章设计了一种对 n 个 $m^e(A)$ 进行 $n-1$ 次合成的算法，此算法可以大大减少运算量，具体算法如下。

算法 2.1

$m_0^e(A)\leftarrow m^e(A)$

$m(A)\leftarrow m_0^e(A)$

$l\leftarrow n$

$m\leftarrow n-1$

$\text{flag}_{m-1}\leftarrow 0$

while $l>1$

do $m_m^e(A)\leftarrow m_{m-1}^e(A)\oplus m_{m-1}^e(A)$

$\text{flag}_m\leftarrow l\%2$

$l\leftarrow \text{int}(l/2)$

if $\text{flag}_{m-1}=1$

then $m(A)\leftarrow m(A)\oplus m_{m-1}^e(A)$

end

$m(A)\leftarrow m(A)\oplus m_m^e(A)$

2.2.3 $G\text{-}G_h$ 合成规则的评价

本节用实例验证的方法说明 $G\text{-}G_h$ 合成规则的优越性。

例 2.1　假设有识别框架 $\Theta=\{A,B\}$，4 个证据的基本信任分配函数 m_1、m_2、m_3、m_4 分别为：$m_1=(A,B,\Theta)=(0.9,0.1,0)$，$m_2=(A,B,\Theta)=(0,1,0)$，$m_3=(A,B,\Theta)=(1,0,0)$，$m_4=(A,B,\Theta)=(0.7,0.3,0)$，分别用 D-S 合成规则、Smets 合成规则、Yager 合成规则、Lefevre 合成规则、Murphy 合成规则和 G-G_h 合成规则合成这 4 个证据的结果如表 2.1 所示。

表 2.1　不同合成规则的合成结果表

规则＼结果＼运算	Ⅰ	Ⅱ	Ⅲ	Ⅳ
	$m_1\oplus m_2$	$m_1\oplus m_4$	$m_2\oplus m_3$	$m_1\oplus m_2\oplus m_3\oplus m_4$
D-S 合成规则	(0,1,0)	(0.95,0.05,0)	无法合成	无法合成
Smets 合成规则	(0,0.1,0.9)	(0.63,0.03,0.34)	(0,0,1)	(0,0,1)
Yager 合成规则	(0,0.1,0.9)	(0.63,0.03,0.34)	(0,0,1)	(0.7,0.03,0.27)
Lefevre 合成规则	(0.405,0.595,0)	(0.902,0.098,0)	(0.5,0.5,0)	(0.823,0.177,0)
Murphy 合成规则	(0.4,0.6,0)	(0.94,0.06,0)	(0.5,0.5,0)	(0.964,0.036,0)
G-G_h 合成规则	(0.315,0.44,0.245)	(0.814,0.084,0.102)	(0.375,0.375,0.25)	(0.849,0.08,0.071)

1. G-G_h 合成规则处理高冲突证据的合理性

运算Ⅰ和Ⅲ是两个高冲突证据合成的例子，以上 6 种合成规则在处理这两个运算时的效果有所不同。

D-S 合成规则在处理这两个运算的时候都遇到了困难。D-S 规则处理运算Ⅰ的结果为完全相信 B，而把证据 1 完全忽略了，其结果有悖人类正常推理思维。D-S 规则处理运算Ⅲ时，遇到了不能合成的情况，因为运算Ⅲ的两个证据完全冲突，即冲突系数 $k=1$，导致归一化系数 $1/(1-k)$ 不存在。

Smets 合成规则和 Yager 合成规则在处理这两个运算的时候得到了相同的结果，但是结果不合理。这两种合成规则把冲突完全分配给识别框架 Θ，导致 $m(\Theta)$ 的值很大，而 $m(A)$ 和 $m(B)$ 的值很小，使得合成结果无法支持决策，失去了证据合成的意义。

Lefevre 合成规则、Murphy 合成规则和 G-G_h 合成规则在处理这两个运算时获得了较好的合成结果。这 3 种合成规则获得的结果都表达了对 A、B 持有几乎同等的信任程度；不同的是，G-G_h 合成规则在表达这一观点的同时也表达了不确定性，将一部分冲突分配给识别框架 Θ，从而更准确地反映了两个证据合成的真实情况。

综上所述，在处理高冲突证据合成时，Lefevre 合成规则、Murphy 合成规则和 G-G_h 合成规则都具有可用性，但是 G-G_h 合成规则较其他两个规则更为精确。

2. G-G_h 合成规则处理低冲突证据的合理性

运算Ⅱ是一个低冲突证据合成的例子，以上 6 种合成规则处理这个运算的效果仍然有所不同。

Smets 合成规则和 Yager 合成规则在处理这两个运算的时候得到了相同的结果，但结果依然不太合理。这两种合成方法把冲突完全分配给识别框架 Θ，导致合成结果对 A 的支持度低于原始证据。这两种合成方法由于过于谨慎，导致合成结果存在巨大的不确定性信

息，给决策带来了困难。

其他 4 种合成规则均获得了明显支持 A 的合成结果，结果符合人类正常的推理思维。因此，这 4 种合成规则对于低冲突证据都有效。

3. G-G_h 合成规则处理混合型证据的合理性

混合型证据是指既包括高冲突证据也包括低冲突证据的一组证据，运算Ⅳ反映了混合型证据的特点。以上 6 种合成规则处理运算Ⅳ的效果同样有很大差别。

D-S 合成规则不能处理运算Ⅳ，因为运算Ⅳ中存在完全冲突的证据，导致归一化因子不存在。

Smets 合成规则获得的处理结果为(0,0,1)，因为存在完全冲突的证据，导致该合成规则把所有支持度分配给 Θ，而忽视这 4 个证据中有 3 个是支持 A 的现实。因此，Smets 合成规则在合成混合型证据时可能出现问题。

Murphy 合成规则获得的结果明确支持 A，但是支持度非常接近于 1。导致这一结果的原因是：在把原始证据的基本信任分配函数进行平均后，Murphy 使用 D-S 合成规则对平均后的基本信任分配函数进行合成。由于 D-S 合成规则的归一化运算，导致了合成过程中信任分配快速向支持度较高的焦元集中。

Yager 合成规则、Lefevre 合成规则和 G-G_h 合成规则在合成混合型证据时获得了较好的结果。其中，Yager 合成规则的结果对于 Θ 的信任分配较大，显得比较谨慎；而 Lefevre 合成规则的结果对于 Θ 的信任分配为 0，显得比较激进。

G-G_h 合成规则相较于 Murphy 合成规则而言，由于采用有效冲突合成法替代了 D-S 合成规则做最后的合成，使得信任分配向支持度较高的焦元集中的速度有所降低。同时，G-G_h合成规则将一部分未决的冲突分配给 Θ，等待下一步合成再行裁决，这样的处理更为精确地反映了 D-S 证据理论对于“不确定”和“不知道”的区分和处理。

综上所述，相比于其他 5 个合成算法，G-G_h 合成规则在处理混合型证据的合成问题时表现出了更为突出的优越性。

4. G-G_h 合成规则处理多维证据的合理性

多维证据是一种混合型证据，一组多维证据可能包含高冲突的证据，也可能包含低冲突的证据。前面证明了 G-G_h 合成规则在处理高冲突、低冲突和混合型证据的合成问题时都表现出了其优越性。因此，G-G_h 合成规则完全有能力处理多维证据的合成问题，并能够获得满意的合成结果。

2.3 EBTrust 信任度评估模型

本节基于多维证据以及 G-G_h 合成规则的研究，设计了一种基于多维证据的信任度评估模型，简称为 EBTrust 信任度评估模型，下面详细阐述该模型。

2.3.1 模型框架

EBTrust 信任度评估模型类似于集中式信任度评估模型，其框架结构如图 2.3 所示。

其中，证据采集模块负责原始证据的采集工作，证据形式化处理模块负责把形式多样、

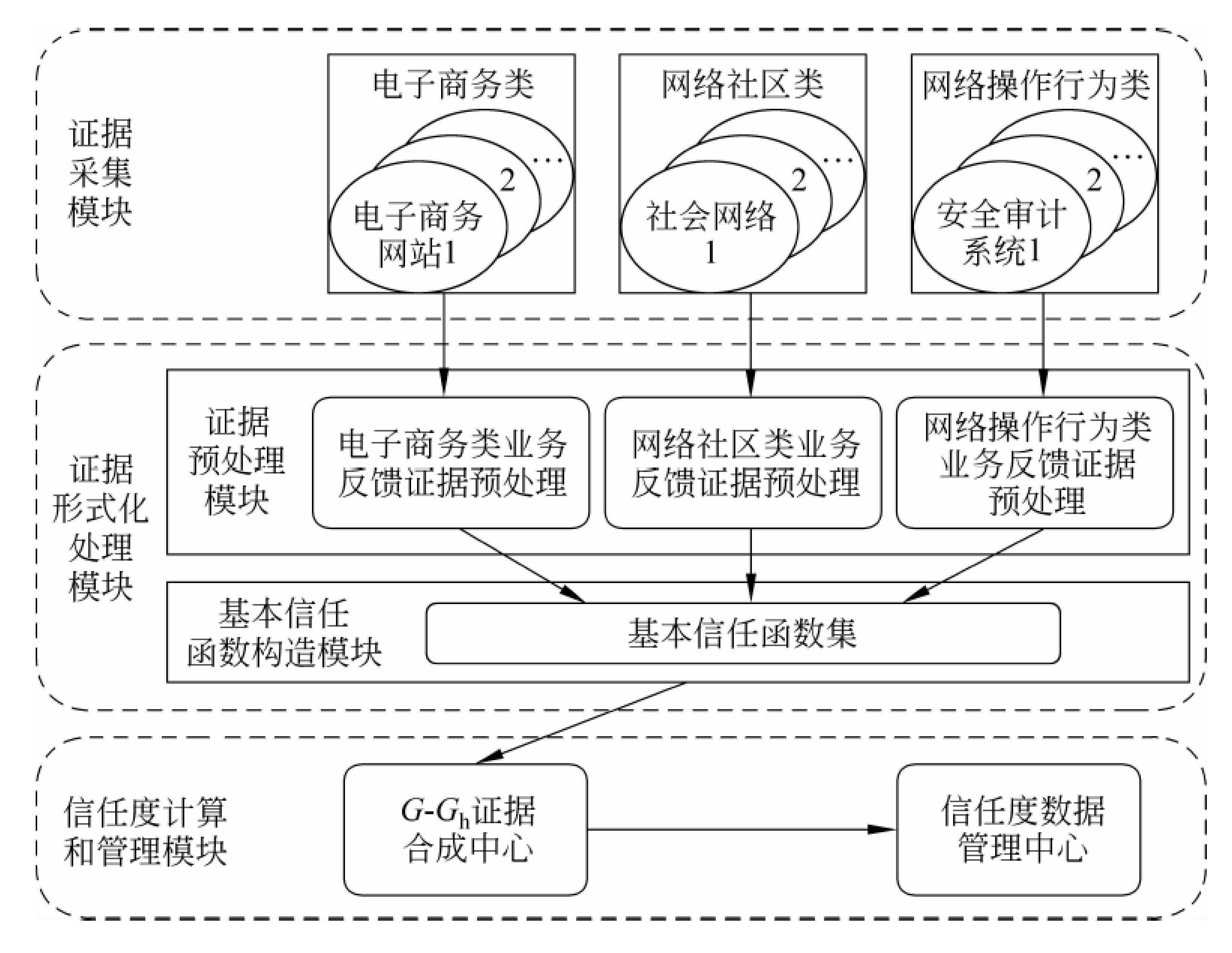


图 2.3　EBTrust 信任度评估模型框架

结构复杂的原始证据进行分类处理并实现形式化表达，信任度计算和管理模块负责信任度的最终计算以及对信任度数据进行存储、更新、检索、发布等管理工作。

2.3.2　证据的采集

EBTrust 信任度评估模型涉及的多维证据包括电子商务类业务反馈证据、网络社区类业务反馈证据和网络操作行为类业务反馈证据 3 类，可以分别在相应的应用系统中获得。

电子商务类业务反馈证据可以从各个电子商务网站的服务器获得，网络社区类业务反馈证据可以从各个网络社区的服务器获得，网络操作行为类业务反馈证据可以从各个安全审计系统中获得。由于本章侧重于模型的设计，而非模型的具体实现，因此本章对这一部分内容不做更为的深入介绍。

2.3.3　证据的形式化处理

对证据进行形式化处理的目的是将用自然语言或者其他非数学语言表达的结构复杂的多维证据转变为用数学语言表达的可以直接参与计算的形式。本章将证据的形式化处理过程分为两个步骤进行，即证据预处理和基本信任函数的构造。

1. 证据的预处理

从证据源获得的原始证据多为自然语言或者其他非数学语言表达的证据，且来自不同证据源的同一类证据可能具有不同的复杂结构，因此对原始证据作必要的预处理显得十分重要。原始证据预处理的目标是，把结构复杂多样的自然语言表达的证据转化为第 2 章 2.1 节设计的数据结构，即把电子商务类业务反馈证据、网络社区类业务反馈证据和网络操作行为类业务反馈证据分别转化为以 Evi(Cla，Ide，T，Val，Res，Asse)、Evi (Cla，Eve，T，Disti)和 Evi (Cla，T，Lev)形式表达的证据。

2. 电子商务类业务反馈证据的预处理

对于任意一条电子商务类业务反馈证据进行预处理后的数据结构为 Evi(Cla,Ide,T,Val,Res,Asse),并且规定:

(1) 证据类型变量 Cla=1。

(2) 网络主体身份变量 Ide∈{−1,1},且 Ide=1 表明当前主体为卖家,Ide=−1 表明当前主体为买家。

(3) 时间变量 T 等于当前证据产生的时间,即当前在线交易发生的时间。

(4) 交易价值变量 Val 等于当前证据指向的交易业务标的物的价值,用货币计量。

(5) 交易结果变量 Res∈{−1,0,1},且 Res=−1 表明当前交易失败且责任在己方,Res=0 表明当前交易失败且责任在对方,Res=1 时表明当前交易成功完成。

(6) 交易评价变量 Asse∈{−1,0,1},且 Asse=−1 表明对方给予己方负面评价,Asse=0 表明对方给予中性评价或者未作评价,Asse=1 表明对方给予正面评价。

电子商务类业务反馈证据的预处理模块负责对获得的电子商务类业务反馈证据进行上述处理,处理后的电子商务类业务反馈证据的形式如下:

```
Evi(1,1,2011.10.1,98,1,1)
```

这一条证据的含义是:该证据指向的网络主体于 2011 年 10 月 1 日作为卖家进行了一次在线交易,货物价值为 98 元,交易成功并获得了正面评价。

3. 网络社区类业务反馈证据的预处理

对于任意一条网络社区类业务反馈证据进行预处理后的数据结构为 Evi(Cla,Eve,T,Disti),并且规定:

(1) 证据类型变量 Cla=2。

(2) 事件变量 Eve∈{1,2,3,4},且 Eve=1 表明当前证据指向的网络主体的主页被浏览,Eve=2 表明原发帖被浏览,Eve=3 表明被管理员删帖,Eve=4 表明被管理员禁止发帖。

(3) 时间变量 T 等于当前证据产生的时间,即当前事件发生的时间。

(4) 事件判定变量 Disti 的值与 Eve 相关联,具体取值如表 2.2 所示。

表 2.2 Disti 与 Eve 对照的含义

Eve	Disti
1	被浏览的次数
2	被浏览的次数
3	Disti=1,对象为发帖者;Disti=−1,对象为跟帖者
4	Disti=1,对象被暂时禁止发帖;Disti=−1,对象被永久禁止发帖或被封号

网络社区类业务反馈证据的预处理模块负责对获得的网络社区类业务反馈证据进行上述处理,处理后的网络社区类业务反馈证据的形式如下:

```
Evi(2,1,2010.10.1,4)
```

这一条证据的含义是该证据指向的网络主体在某个网络社区的主页于 2010 年 10 月 1 号

累计被浏览超过 4000 次，但低于 5000 次；

```
Evi(2,4,2010.10.1,- 1)
```

这一条证据的含义是 2010 年 10 月 1 日该证据指向的网络主体在某网络社区被无限期禁止发帖或者被管理员封号。

4. 网络操作行为证据的预处理

对于任意一条网络操作行为证据进行预处理后的数据结构为 Evi（Cla，T，Lev），并且规定：

（1）证据类型变量 Cla=3。

（2）时间变量 T 等于当前证据所指向的行为发生的时间。

（3）行为危害级别（或称攻击级别）Lev∈{1，2，3，4，5}，攻击级别分类如表 2.3 所示[1]。

表 2.3　危害级别分类

攻击级别	攻击类型	攻击行为列举
1	信息泄露	读取文件、内存数据、注册表和进程，探测 IP 地址、软件版本、操作系统指纹、端口和键盘记录等
2	拒绝服务	应用或进程出错、CPU 消耗、内存消耗、系统或设备出错、进程资源消耗、网络带宽资源消耗、服务质量下降、磁盘空间消耗等
3	数据破坏与欺骗	篡改文件系统、内存数据、操作系统账户、口令或者密码、系统内核、路由、数据库配置，以及端口重定向等
4	入侵控制	非法执行程序、非法提升操作系统权限、非法提升数据库权限、非法利用资源、非法开启后门、越权访问文件系统、非法获得 Shell 等
5	对抗性	绕过病毒检测、穿透防火墙、绕过垃圾邮件检测、躲避 IDS/IPS、静态隐藏功能、运行隐藏功能、通信隐藏功能等

网络操作行为证据的预处理模块负责对获得的网络操作行为证据进行上述处理，处理后的网络操作行为证据的形式如下：

```
Evi(3,2010.10.1,3)
```

这一条证据的含义是该证据指向的网络主体于 2011 年 10 月 1 日做出了数据破坏与欺骗类的网络攻击行为，且该行为危害级别被定为 3。

2.3.4　基本信任函数的构造

1. 识别框架的构造和焦元的确定

对于任意一个网络主体，假设证据十分充分，那么对其信任情况的判定只有两种，即信任和不信任，据此可以构造判定某网络主体信任情况的识别框架，即

$$\Theta = \{信任,不信任\}$$

简写为

$$\Theta = \{t, d\} \tag{2.21}$$

因此

$$2^{\Theta} = \{\varnothing, \{t\}, \{d\}, \Theta\} \tag{2.22}$$

对于任意一条证据，由基本信任分配函数的定义可知 $m(\varnothing)=0$；$\{t\}$描述了对当前主体判定为信任，$m(\{t\})$表达了当前证据对于该判定的支持程度；$\{d\}$描述了对当前主体判定为不信任，$m(\{d\})$表达了当前证据对于该判定的支持程度；Θ 描述了当前证据无法判断当前主体的信任情况，$m(\Theta)$表达了当前证据对于该判定的支持程度。因此，对于所有证据，Θ 的可能的焦元有$\{t\}$、$\{d\}$、Θ，本章分别将这些焦元表示为命题 T、D、Θ，即

$$T=\{t\},\quad D=\{d\},\Theta$$

因此，本章构造的基本信任分配函数的基本形式为

$$m(T,D,\Theta) \tag{2.23}$$

2. 电子商务类业务反馈证据的基本信任函数构造

本章根据实际电子商务中交易结果和评价对于信任度的影响来构造电子商务类业务反馈证据的基本信任函数，对于网络主体 A，不论他是卖家还是买家，具体的构造方法均如表 2.4 所示。

表 2.4　电子商务类业务反馈证据 mass 函数

序号	Evi(Cla,Ide,T,Val,Res,Asse)	$m(T,D,\Theta)$	说　明
1	(1,Ide,T,Val,1,1)	(1,0,0)	交易成功完成，且获得正面评价，认为 A 可信
2	(1,Ide,T,Val,1,0)	(0.5,0,0.5)	交易成功完成，且获得中性评价或者评价缺失，认为 A 被不完全信任
3	(1,Ide,T,Val,1,−1)	(0,1,0)	交易成功完成，但获得负面评价，认为 A 不可信
4	(1,Ide,T,Val,0,1)	(0.5,0,0.5)	交易因对方原因而没有完成，且获得正面评价，认为 A 被不完全信任
5	(1,Ide,T,Val,0,0)	(0,0,1)	交易因对方原因而没有完成，且获得中性评价或评价缺失，认为无法判断 A 是否可信
6	(1,Ide,T,Val,0,−1)	(0,1,0)	交易因对方原因而没有完成，且获得负面评价，认为 A 不可信
7	(1,Ide,T,Val,−1,1)	(0.5,0,0.5)	交易因 A 的原因而没有完成，但获得正面评价，认为 A 被不完全信任
8	(1,Ide,T,Val,−1,0)	(0,0.5,0.5)	交易因 A 的原因而没有完成，且获得中性评价或评价缺失，认为 A 被不完全信任
9	(1,Ide,T,Val,−1,−1)	(0,1,0)	交易因 A 的原因而没有完成，且获得负面评价，认为 A 不可信

其中，证据 Evi(Cla,Ide,T,Val,Res,Asse) 中的变量 Cla、Ide、T 和 Val 用作计算证据的权重，详细计算方法将在 2.3.5 节论述。

3. 网络社区类业务反馈证据的基本信任函数构造

本章根据实际网络社区中主体的行为对其信任度的影响来构造网络社区类业务反馈证

据的基本信任函数。对于网络主体 A，具体的构造方法如表 2.5 所示。

表 2.5　网络社区类业务反馈证据的 mass 函数

序号	Evi(Cla，Eve，T，Disti)	$m(T,D,\Theta)$	说　明
1	(2，1，T，Disti)	(1，0，0)	A 的主页被浏览，认为 A 可信
2	(2，2，T，Disti)	(1，0，0)	A 发表的帖子被浏览，认为 A 可信
3	(2，3，T，Disti)	(0，1，0)	A 发表的帖子被管理员删除，认为 A 不可信
4	(2，4，T，Disti)	(0，1，0)	A 被管理员禁止发帖，认为 A 不可信

其中，证据 Evi(Cla，Eve，T，Disti) 中的变量 Cla、Eve、T 和 Disti 用作计算证据的权重，详细计算方法将在 2.3.5 节论述。

4. 网络操作行为类业务反馈证据的基本信任函数构造

本章根据实际网络操作行为对网络主体的信任度的影响来构造网络操作行为类业务反馈证据的基本信任函数。由于正常的网络操作行为占绝大多数，所以本章只关注危害性网络操作行为，并将其视为对主体信任度产生衰减作用的因素。对于网络主体 A，具体的构造方法如表 2.6 所示。

表 2.6　网络社区类业务反馈证据的 mass 函数

序号	Evi(Cla，T，Lev)	$m(T,D,\Theta)$	说　明
1	(3，T，Lev)	(0，1，0)	只要 A 产生了网络操作行为证据，则说明 A 做出了危害性网络操作行为，即认为 A 不可信

其中，证据 Evi(Cla，T，Lev) 中的变量 Cla、T 和 Lev 用作计算证据的权重，详细计算方法将在 2.3.5 节论述。

2.3.5　证据权重的计算与处理

我们认为不同类型的证据与同一类型的不同证据对于主体信任度的影响程度是不同的。因此，在信任度评估时，根据各个证据的特征对其赋予一个权重是十分必要的。本节对多维证据的权重计算与处理进行说明。

1. 证据权重的计算

我们设计的信任度更新规则采用定期更新的方法，因此，本章将证据的权重随时间衰减的特性延迟到信任度更新规则的设计中。对于任意一条证据，其权重 W(weight)计算方法如下。

(1) 当 Cla＝1 时，由 2.3.3 节可知此证据为电子商务类业务反馈证据，其权重计算如表 2.7 所示。

表 2.7　电子商务类业务反馈证据的权重计算

Cla	Ide	Val	*W*
1	1	Val≤1000	Val/100
		1000＜Val≤10000	10＋Val/1000
		10000＜Val≤100000	20＋Val/10000
	－1	Val≤100	(Val/100)/4
		100＜Val≤1000	(10＋Val/1000)/4
		1000＜Val≤10000	(20＋Val/10000)/4

（2）当 Cla＝2 时，可知证据为网络社区类业务反馈证据，其权重计算如表 2.8 所示。

表 2.8　网络社区类业务反馈证据的权重计算

Cla	Eve	Disti	*W*
2	1	Disti＜1000	0
		1000＜Disti≤10000	Disti/1000
		10000＜Disti≤100000	10＋Disti/10000
		100000＜Disti≤1000000	20＋Disti/100000
		1000000＜Disti≤10000000	30＋Disti/1000000
		10000000＜Disti≤100000000	40＋Disti/10000000
	2	Disti＜100	0
		100＜Disti≤1000	Disti/100
		1000＜Disti≤10000	10＋Disti/1000
		10000＜Disti≤100000	20＋Disti/10000
		100000＜Disti≤1000000	30＋Disti/100000
		1000000＜Disti≤10000000	40＋Disti/1000000
		10000000＜Disti≤100000000	50＋Disti/10000000
	3	1	10
		－1	5
	4	1	20
		－1	40

（3）当 Cla＝3 时，可知此证据为网络操作行为类业务反馈证据，其权重计算如表 2.9 所示。

表 2.9　网络操作行为证据的权重计算

Cla	Lev	W
3	1	50
	2	60
	3	70
	4	80
	5	90

2. 证据权重的处理

证据权重的处理是指将证据的权重反映到证据合成过程中的方法。基于 $G\text{-}G_h$ 合成规则的特点，本章在构造相似矩阵时引入证据的权重值 W，具体做法如下：

(1) 在基本信任函数构造完成并计算出各个证据的权重值 W 后，将相同信任函数的权重值相加，得出某一基本信任分配函数的权重。

(2) 运用进一法将各个基本信任分配函数的权重值转变为整数，记为 w。

(3) 将权重值 w 作为基本信任分配函数的个数，并对 w 个信任分配函数进行加权平均。

下面举例说明以上步骤。假设证据 E_1、E_2、E_3 的基本信任分配函数分别为(1,0,0)、(0,1,0)和(1,0,0)，证据的权重 W 分别为 5、3.5 和 2，则有信任分配函数(1,0,0)和(0,1,0)的权重分别为 7 和 4。至此，在对基本信任分配函数进行加权时只需对 7 个(1,0,0)和 4 个(0,1,0)进行加权平均即可。

需要说明的是，使用 $G\text{-}G_h$ 合成规则对加权平均后的基本信任分配函数进行合成的次数 $n-1$ 中的 n 是指证据的个数，而不是参与加权平均的基本信任分配函数的个数。

2.3.6　信任度的计算和管理

信任度的计算和管理由基于证据的信任度评估模型的信任度计算和管理模块来完成，该模块包括 $G\text{-}G_h$ 证据合成中心和信任度数据管理中心两个子模块。本节分别介绍这两个模块的功能和算法。

1. 信任度的计算

信任度计算是 $G\text{-}G_h$ 证据合成中心的核心任务。按照前面章节所介绍的方法，可以把多维证据转变为若干个形如(1,0,0)的数组并标记它们的权重，此时便可以利用 $G\text{-}G_h$ 合成规则进行证据合成，其步骤如下：

(1)计算各个基本信任分配函数的总权重，并按照 2.3.4 节的方法获得待加权平均的 x 个基本信任分配函数，x 等于所有基本信任分配函数的总权重，即参与加权平均的基本信任分配函数的个数。

(2) 运用式(2.10)计算证据间的两两冲突强度 G，记录为

$$G_{ij}\quad (i,j=1,2,\cdots,x)$$

(3) 构造形如式(2.11)的相似矩阵，由于本章对证据的处理方法特殊，所以相似矩阵中的相似系数只会出现 0、0.5 和 1。

(4) 运用式(2.12)～式(2.14)计算加强平均后的基本信任分配函数，即

$$m^e(T,D,\Theta) = (m^3(T),m^e(D),m^e(\Theta))$$

(5) 运用式(2.19)和式(2.20)对 m^e 合成 n 次，n 为证据的个数。为减少运算量，本章采用算法 2.1 计算 $m(A)$。

(6) 根据 $m(A)$计算最终信任度 Tru，计算方法如下：

$$\text{Tru} = (\text{Bel}(T),\text{Pl}(T)) \tag{2.24}$$

2. 信任度数据的管理

信任度数据的管理包括信任度的存储、更新、检索和发布等，由于本章的内容不涉及信任度评估系统的具体实现，因此，本节只对信任度的存储和更新做简要介绍。

本章设计的信任度的存储结构为(t,Tru)。其中，t 表示时间，表达了当前信任度值产生的时间；信任度值为上面的计算方法的计算结果 Tru，其形式为一个二元数组，即 $(\text{Bel}(T),\text{Pl}(T))$。

本章设计的信任度评估模型采取定期评估的方式对信任度值周期性地进行更新，并规定 t 时刻的信任度值记为 Tru_t。信任度的更新算法如下：

(1) 获取当期($t-1$ 到 t 时间段内)证据，运用上面所述算法计算当期基本信任分配函数 $\Delta m(A)$。

(2) 取上一期基本信任分配函数 $m(A)$，记为 $m_{t-1}(A)$。

(3) 计算当前时刻基本信任分配函数 $m_t(A)=m_{t-1}(A)\oplus\Delta m(A)$，合成时 $m_{t-1}(A)$和 $\Delta m(A)$的权重分别为 1 和 2。

(4) 运用式(2.24)计算信任度 Tru_t。

2.4 EBTrust 信任度评估模型的实验分析

我们使用 Java 语言开发实现了 EBTrust 信任度评估模型的核心功能模块信任度计算和管理模块，并以此为基础，通过实验分析说明了 EBTrust 信任度评估模型具有抵制共谋和恶意评价行为的功能。

2.4.1 信任度计算和管理模块的设计与实现

1. 功能模块设计

EBTrust 信任度评估模型核心功能模块涉及两个角色，即管理员和普通用户。普通用户既是信任度评估的对象也是信任度评估结果的使用者。管理员是系统的管理者，负责系统的维护。普通用户在系统中的操作主要有用户注册、用户登录和信任度查询。管理员在系统中的操作主要有登录、管理普通用户、管理信任度计算模块和管理信任度管理模块。两种角色在系统中的具体操作如图 2.4 所示。

2. 数据库设计和实现

本章采用 MySQL 5.5 数据库作为 EBTrust 核心功能模块的数据库。创建的数据库名称为 EBTrusteval，各个数据表及其结构如下。

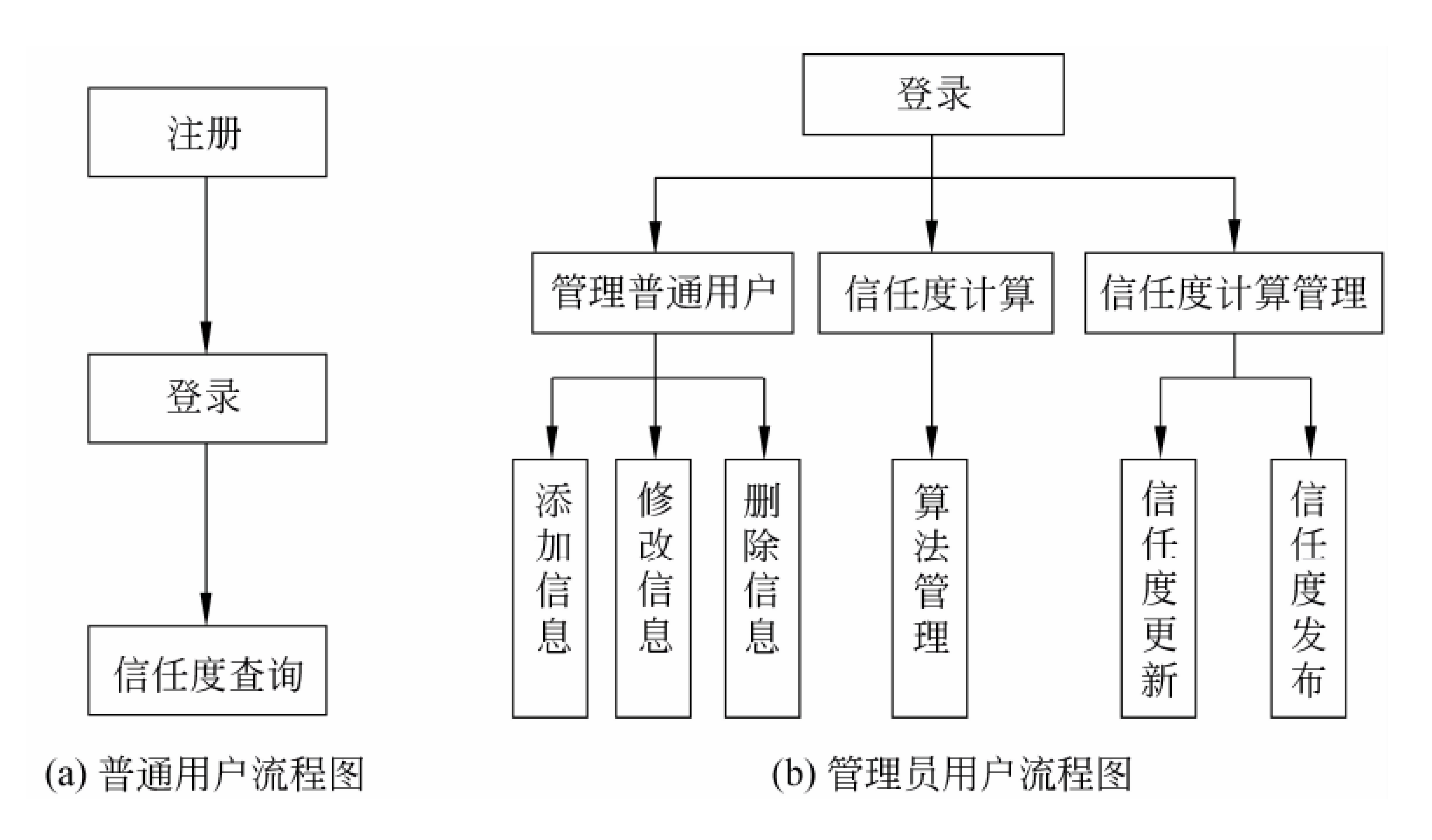


图 2.4　EBTrust 系统流程

1）管理员信息表

本系统的管理信息表 admin 的结构如表 2.10 所示。

表 2.10　表 admin 的结构

字段名称	含　　义	类　　型	约束
id	管理员 id	int	主键
name	管理员名称	varchar(255)	非空
password	管理员密码	varchar(255)	可以为空

2）普通用户信息表

本系统的评估对象同时也是用户，任意一个用户在被纳为评估对象时，系统将自动为其分配 id 和 name。用户只需依据自己的实际身份信息在系统中注册之后便能完成普通用户的所有操作。普通用户信息表 user 的结构如表 2.11 所示。

表 2.11　表 user 的结构

字段名称	含　　义	类　　型	约束
id	普通用户 id	int	主键
name	普通用户名称	varchar(255)	非空
password	普通用户密码	varchar(255)	可以为空

3）mass 表

mass 表用于存放经过证据的形式化处理模块处理之后的基本信任分配函数。mass 表的结构如表 2.12 所示。

表 2.12　表 mass 的结构

字段名称	含　　义	类　　型	约束
id	mass 函数的 id	int	主键
userid	普通用户的 id	int	外键
time	mass 函数对应的证据产生的时间	datetime	非空
t	mass 函数 $m(T)$的值	double	非空

续表

字段名称	含　　义	类　　型	约束
d	mass 函数 $m(D)$的值	double	非空
u	mass 函数 $m(\Theta)$的值	double	非空
weight	当前证据的权重	double	非空

4）trustDegree 表

trustDegree 表用于存放计算后的信任度值。trustDegree 表的结构如表 2.13 所示。

表 2.13　表 trustDegree 的结构

字段名称	含　　义	类　　型	约束
id	信任度的 id	int	主键
time	信任度值产生的时间	datetime	非空
bel	信任函数值，即最低信任度值	double	非空
pl	似然函数值，即最高信任度值	double	非空

5）实体的关系图

根据数据库的建表情况，可以确定本数据库的实体包括管理员（admin）、普通用户（user）、基本信任分配函数（mass）和信任度值（trustDegree），它们的关系如图 2.5 所示。

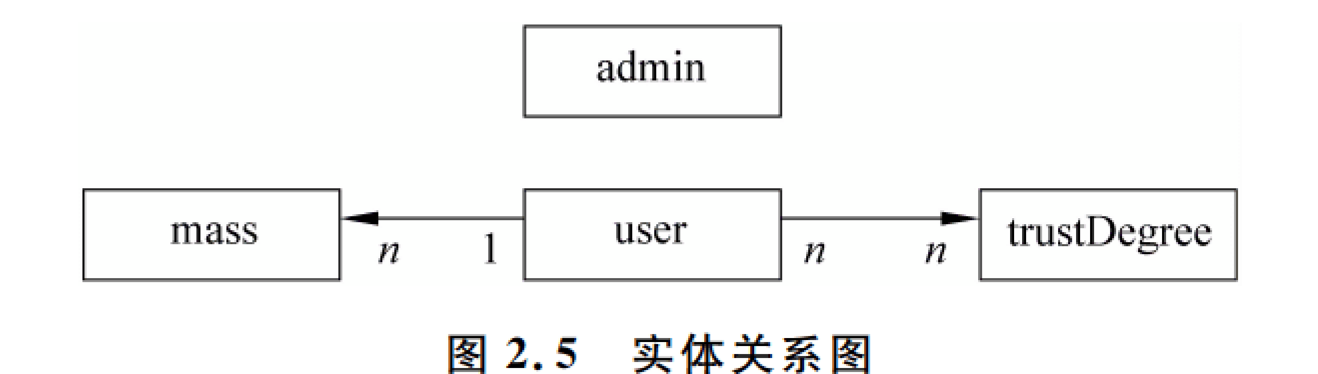


图 2.5　实体关系图

3. 信任度计算功能的实现

信任度计算功能是 EBTrust 信任度评估模型的核心功能，本章基于 Java 语言实现了该功能，具体实现方法如下。

基于 Java 语言面向对象的程序设计思想，本章将基本信任分配函数定义为一个类，即 Mass 类，在证据合成过程中，实际上是对 Mass 类的对象进行运算和操作。Mass 类的基本代码模式如下：

```
double [] m=new double[3];
public void setMass(double x,double y,double z){
m[0]=x;m[1]=y;m[2]=z;}
```

证据合成实质上是对基本信任分配函数的合成，即对于 Mass 类的对象的合成。$G\text{-}G_h$ 合成规则的最后一步是基于有效冲突合成法对加权平均后的基本信任度分配函数进行合成，而最基本的合成过程是对两个加权平均后的基本信任分配函数做一次合成，其方法为

```
public Mass merge(Mass x,Mass y) {…}
```

其关键计算方法如下：

(1) 有效冲突 G_h 的计算方法

```
double k=Math.round((m1.m[0] * m2.m[1]+m1.m[1] * m2.m[0]) * 10000)/10000.0;
double h=Math.round((m1.m[0] * m2.m[0]+m1.m[1] * m2.m[1]+m1.m[2] * m2.m[2]) *
10000)/10000.0;
double gh=Math.round(h/(k+h) * 10000)/10000.0;
```

(2) 合成后的基本信任分配函数的计算方法如下

```
m3.m[0]=Math.round((m1.m[0] * m2.m[0]+m1.m[0] * m2.m[2]+m1.m[2] * m2.m[0]+
(m1.m[0]+m2.m[0])/(m1.m[0]+m2.m[0]+m1.m[1]+m2.m[1]) * gh * k) * 10000)/10000.0;
m3.m[1]=Math.round((m1.m[1] * m2.m[1]+m1.m[1] * m2.m[2]+m1.m[2] * m2.m[1]+
(m1.m[1]+m2.m[1])/(m1.m[0]+m2.m[0]+m1.m[1]+m2.m[1]) * gh * k) * 10000)/10000.0;
m3.m[2]=Math.round((m1.m[2] * m2.m[2]+(1-gh) * k) * 10000)/10000.0;
```

G-G_h 合成规则最终需要对加权平均后的基本信任分配函数做 $n-1$ 次合成,基于算法 2.1 实现的算法为

```
public class MergeofSeveral(int x,Mass y) {…}
```

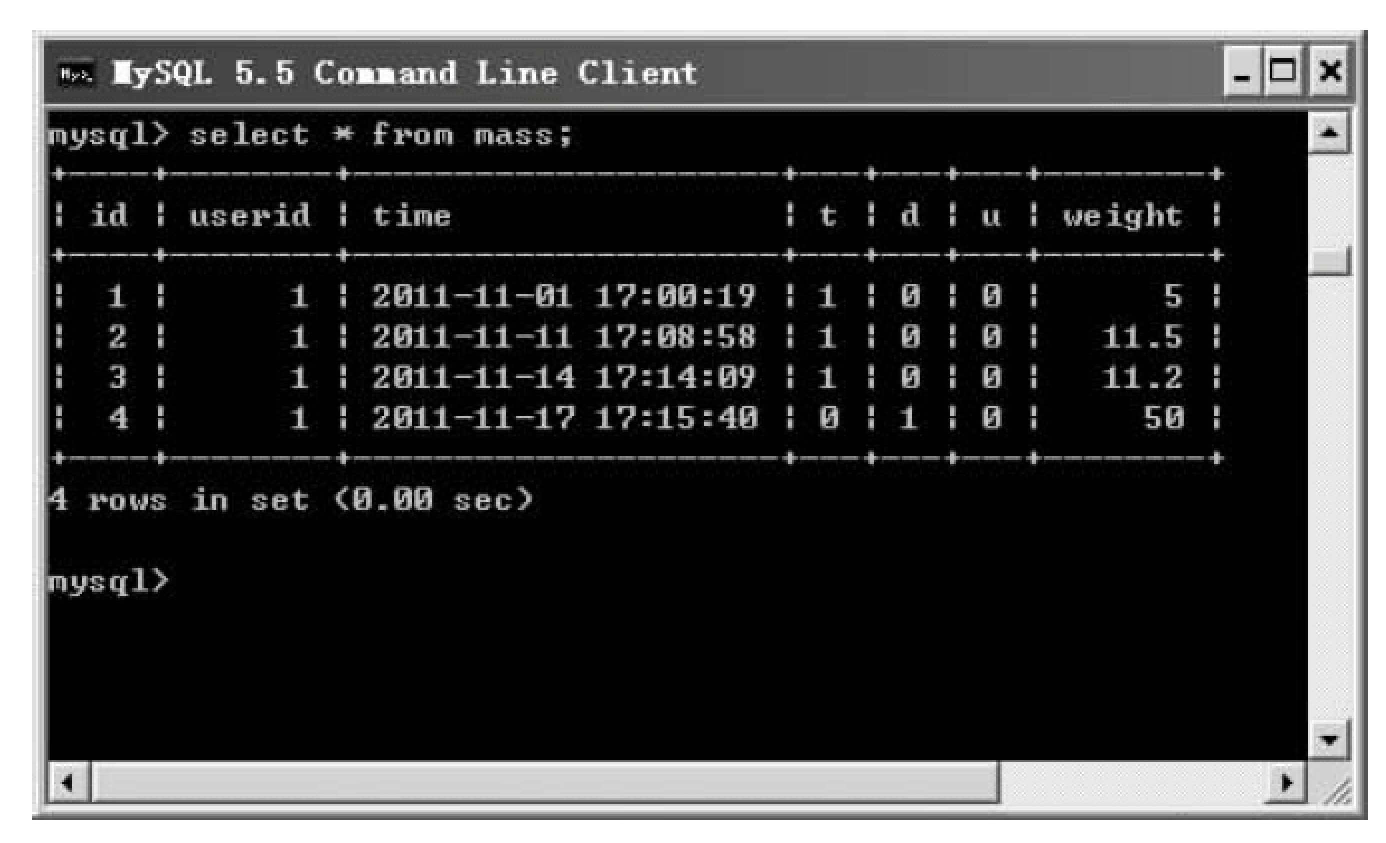
最终信任度计算只需调用 MergeofSeveral 方法即可。

2.4.2 实验分析

1. 证据权重的作用分析

EBTrust 信任度评估模型区别对待每一个证据,认为不同的证据对于主体信任度的影响程度是不一样的,因此本章为每一个证据赋一个权重,并在信任度计算算法中体现这个权重。本节的实验用于演示并分析证据权重的作用。

实验 2.1 假设在时间窗口 t 内,系统搜集到关于主体 A 的证据有 4 条,分别为两条电子商务类业务反馈证据、一条网络社区类业务反馈证据以及一条网络操作行为证据,形式化表达为:Evi1(1,1,t,500,1,1)、Evi2(1,1,t,1500,1,1)、Evi3 (2,1,t,12000)和 Evi4 (3,t,1)。这 4 条证据的基本信任分配函数及其权重在 MySQL 数据库中的 mass 表中的记录如图 2.6 所示。



MySQL 5.5 Command Line Client

mysql> select * from mass;

id	userid	time	t	d	u	weight
1	1	2011-11-01 17:00:19	1	0	0	5
2	1	2011-11-11 17:08:58	1	0	0	11.5
3	1	2011-11-14 17:14:09	1	0	0	11.2
4	1	2011-11-17 17:15:40	0	1	0	50

4 rows in set (0.00 sec)

mysql>

图 2.6 实验 2.1 的 mass 表

如果不考虑证据的权重，那么信任度计算的结果为(0.9933,0.9966)，表明主体 A 的最小信任度为 0.9933，最大信任度为 0.9966；如果考虑证据的权重，那么信任度计算的结果为(0.0566,0.1091)，表明主体 A 的最小信任度为 0.0566，最大信任度为 0.1091。证据 4 是网络操作行为证据，产生此类证据说明主体 A 做了危害网络安全的行为，应该使他的信任度受到惩罚性的衰减，而证据的权重正好体现了这一点。

证据的权重将证据对于信任度的影响进行分级，在信任度计算中区别对待不同的证据，这符合现实人类社会建立信任的实际需求。

2. 对于共谋行为的抵制

现有的信任度评估模型缺乏抵制共谋行为的有效措施，使得有些人可以通过共谋行为来提高彼此的信任度，比如在电子商务系统中，通过相互大量买卖价格极低的商品来提高自己的信任评级。本模型的信任度评估算法使得希望通过共谋行为要达到预期效果的成本大大增加，进而减少共谋行为。

实验 2.2 实验 2.1 中信任度评估的结果为(0.0566,0.1091)，假设主体 A 希望通过共谋行为来提升自己的信任度值，他的方法是在线出售价值 500 元的商品并寻找共谋合伙人来购买该商品。那么他要付出的代价和获得的结果趋势如图 2.7 所示，其中横坐标表示交易总金额，本章以交易总金额来衡量提升信任度的代价。

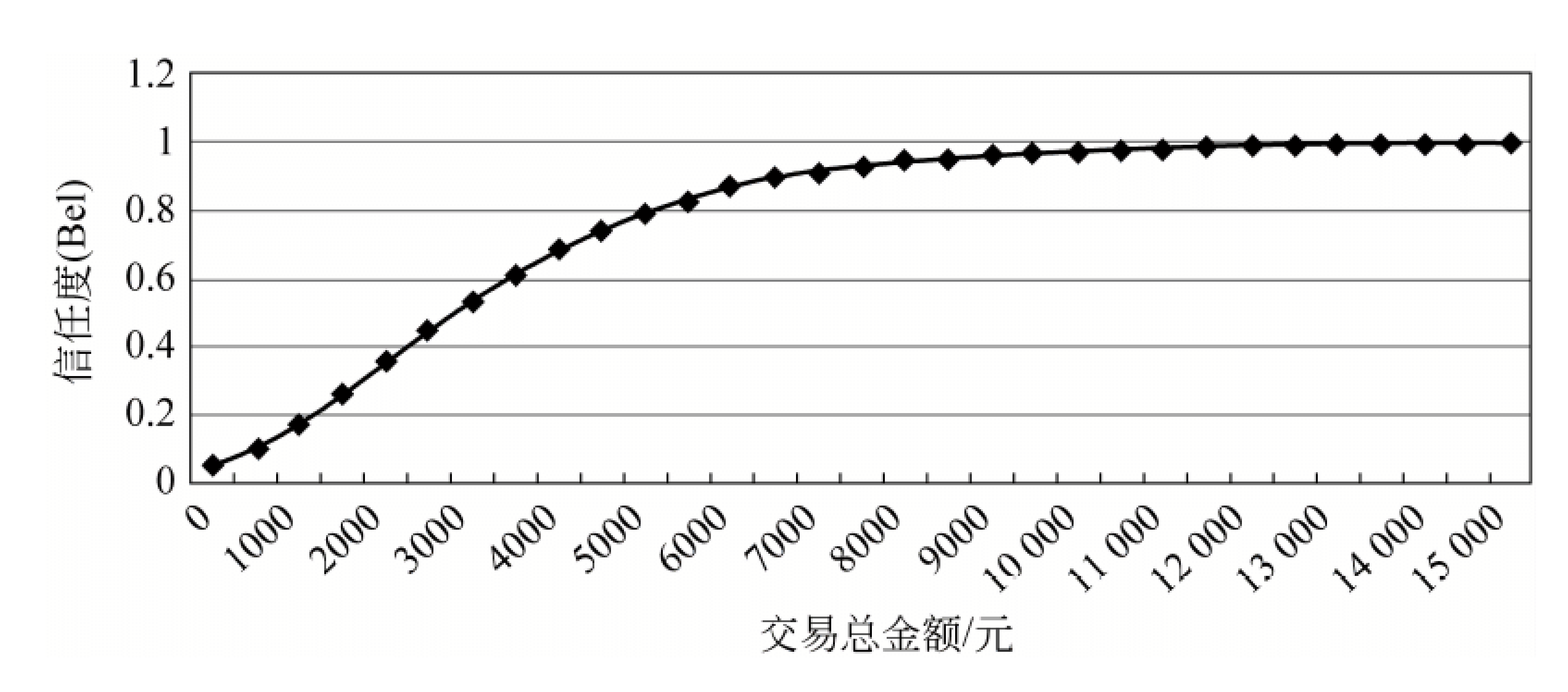


图 2.7 提升信任度的开销趋势

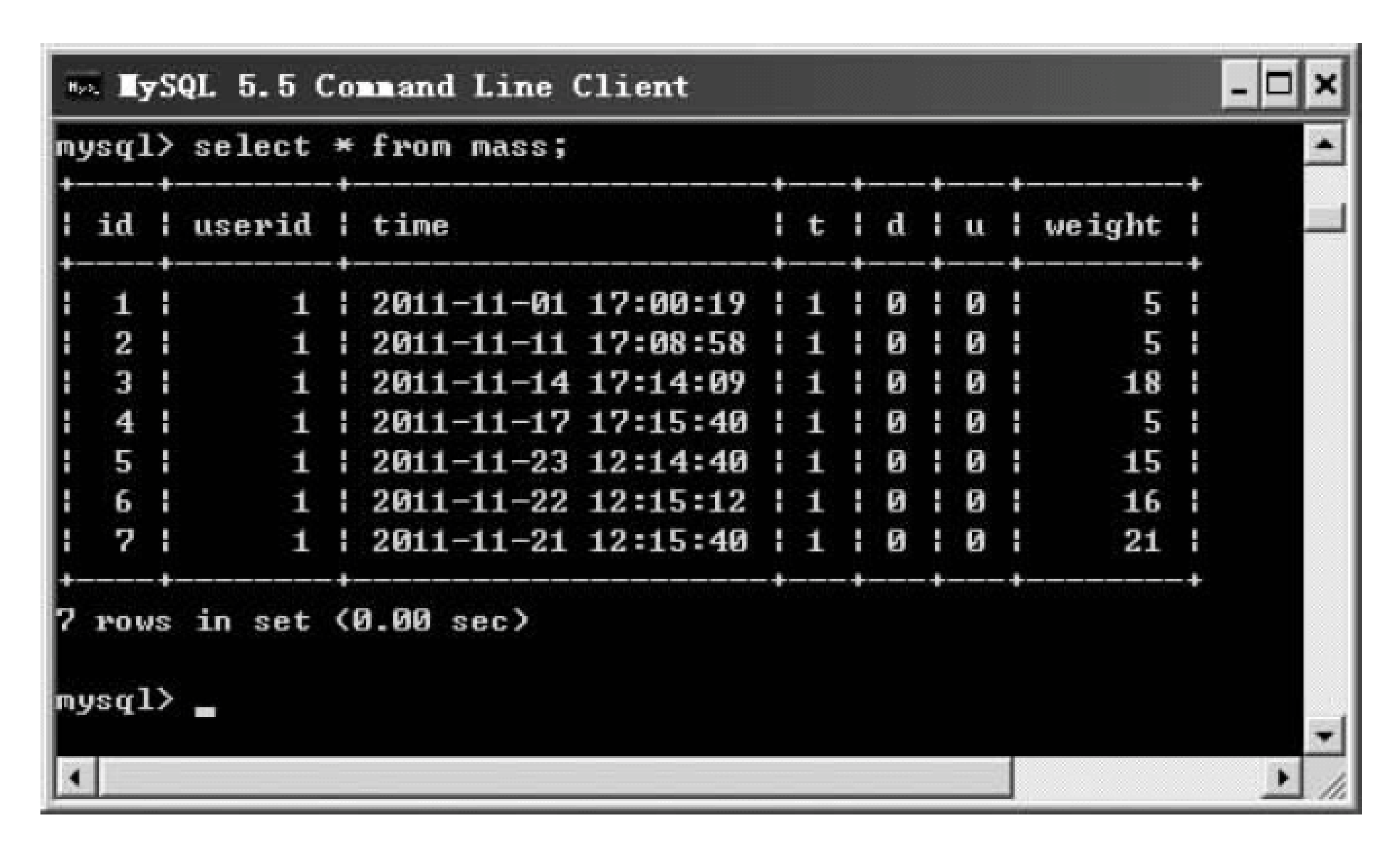
我们选取几个重要的点来考察开销的大小。由图 2.7 可见，将最小信任度(Bel 值)提升到 0.6、0.7、0.8、0.9 和 0.95 的代价分别为 3500 元、4500 元、5500 元、7000 元和 8500 元。由此可见，主体希望通过共谋行为来提升信任度的代价比较大，这样可以有效减少共谋行为对于信任度评估的影响。

3. 对于恶意评价行为的抵制

现有的信任度评估模型缺乏有效的抵制恶意行为的措施，使得某些网络主体可以通过与竞争对手进行交易并给予恶意差评的方法来降低竞争对手的信任度。本模型的信任度评估算法使得希望通过恶意评价行为要达到预期效果的成本大大增加，进而减少恶意评价行为。

实验 2.3 现有主体 A 在时间窗口 t 内产生的证据的基本信任分配函数如图 2.8 所示，其信任度为(1.0,1.0)；而主体 B 希望通过与 A 在线交易并给予恶意差评的方法降低 A 的信任度。假设 A 的在线商店只出售价格为 500 元的商品，那么 B 降低 A 的信任度所需的

开销趋势图如图 2.9 所示。



MySQL 5.5 Command Line Client

mysql> select * from mass;

id	userid	time	t	d	u	weight
1	1	2011-11-01 17:00:19	1	0	0	5
2	1	2011-11-11 17:08:58	1	0	0	5
3	1	2011-11-14 17:14:09	1	0	0	18
4	1	2011-11-17 17:15:40	1	0	0	5
5	1	2011-11-23 12:14:40	1	0	0	15
6	1	2011-11-22 12:15:12	1	0	0	16
7	1	2011-11-21 12:15:40	1	0	0	21

7 rows in set (0.00 sec)

mysql>

图 2.8　实验 2.3 的 mass 表

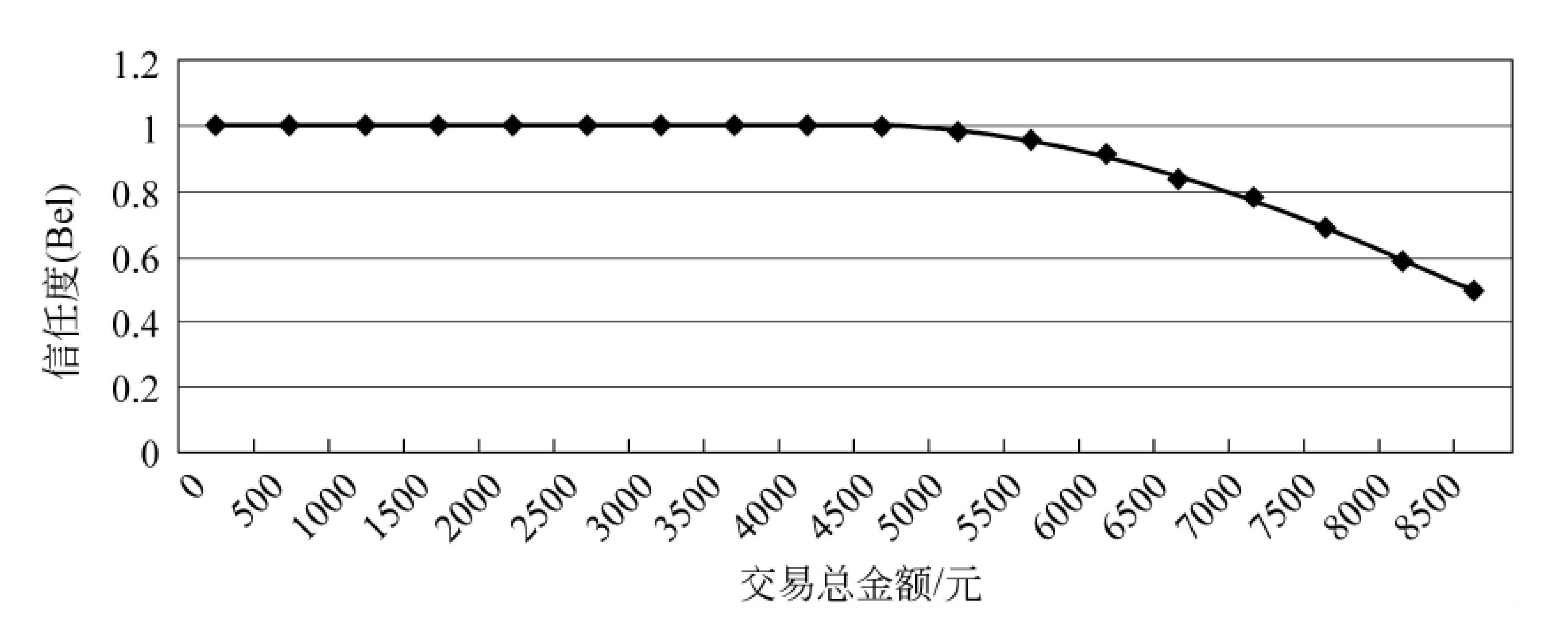


图 2.9　降低竞争对手信任度的开销趋势

由图 2.9 可以看出，当主体 B 累计交易第 10 次时，主体的 A 的信任度才低于 1，而此时 B 的开销是 5000 元；当 B 继续与 A 交易并给予恶意差评时，A 的信任度持续下降；当完成第 15 次交易时，A 的信任度低于 0.5，而此时 B 的开销是 8500 元。由此可见，主体 B 希望通过恶意评价行为来降低主体 A 的信任度的代价比较大，而且当 A 发现 B 的恶意行为后可以拒绝与之继续交易，这样可以有效减少恶意评价行为对于信任度评估的影响。

2.5　本章小结

本章提出多维证据概念来扩展信任度评估的证据，将证据分为电子商务类业务反馈证据、网络社区业务反馈证据和网络操作行为类业务反馈证据。同时针对 D-S 证据理论及其改进展开深入研究，提出了一种基于冲突强度 G 和有效冲突 G_h 的新的证据合成规则，即 G-G_h 合成规则。在此基础上，本章提出了一种基于证据的信任度评估模型 EBTrust，并在理论研究的基础上实现了 EBTrust 的核心功能模块。最后详细阐述了以 EBTrust 的模型结构、证据的形式化处理方法、信任度计算的核心算法和实现了的 EBTrust 的核心模块为平台，通过实验证明了 EBTrust 具有抵制共谋和恶意评价行为的能力。

参 考 文 献

[1] 胡影，郑康峰，杨义先. 网络攻击效果提取和分类[J]. 计算机应用研究，2009，26(3)：1119-1122.

[2] Dempster A P. Upper and Lower Probabilities Induced by a multiplicand Mapping[J]. Annals of Mathematical Statistics，1967，38：325-339.

[3] Dempster A P. A Generalization of Bayesian Inference [J]. Journal of the Royal Statistical Society，1968，Series B 30：205-245.

[4] 杨风暴，王肖霞. D-S 证据理论的冲突证据合成方法[M]. 北京：国防工业出版社，2010. 2-3.

[5] Smets P. The Combination of Evidence in the Transferable Belief Model[J]. IEEE Trans. on Pattern Analysis and Machine Intelligence，1990，12(5)：447-458.

[6] Yager R R. On the Dempster-Shafer Framework and New Combination Rules[J]. IEEE Trans. on System，1989，41(2)：93-137.

[7] Lefevre E，Clot O，et al. A Generic Framework for resolving the conflict in the combination of belief structures[C]. In：The 3rd International Conference on Information Fusion，2000，1998：182-188.

[8] Murphy C K. Combining Belief Functions when Evidence Conflicts [J]. Decision Support System，2000，29(1)：1-9.

[9] 陈文亮. 一种基于证据的信任度评估模型研究. 北京信息科技大学硕士学位论文，2011.

第3章　基于行为检测的信任度评估技术

前面两章已经指出，目前信任度评估所依据的证据通常有两类，即凭证证据和行为证据。凭证证据是指网络主体所拥有的某些数字凭证。行为证据是指可证明网络主体的网络行为的事实或者记录。本章专门讨论基于行为证据的信任度评估技术，且重点讨论基于网络操作行为的信任度评估模型和算法。

3.1　网络行为检测技术

本章所讨论的网络操作行为是指在网络技术层面上表现的行为，包括正常行为和入侵行为。其中入侵行为是指诸如非法访问、口令猜测、DOS攻击和木马攻击等危害网络安全的行为，这些行为与交易相关并可能破坏交易公平性或者危害网络安全。目前利用网络存在的安全脆弱性，黑客针对网络系统的入侵攻击事件越来越频繁地出现。无疑，网络主体实施危害性网络操作行为，将对其信任度产生较为严重的破坏性影响。我们知道，交易反馈信息是普通用户容易知道的，但网络攻击行为却不容易发现，需要借助入侵检测和安全审计等一些特殊的网络行为检测技术手段来识别。

目前网络行为检测主要通过入侵检测技术实现，入侵检测作为一种动态的安全防范技术，能实时发现和识别各种违反安全策略的网络行为，并能做出记录、报警和阻断等响应，提供了一种比较积极主动的网络安全防御手段。下面概要地介绍入侵检测技术的一般知识，包括入侵检测的基本概念、入侵检测系统的功能结构、入侵检测系统的分类和入侵检测的分析方法。

3.1.1　入侵检测的基本概念

入侵(intrusion)指的是任何企图破坏计算机资源的保密性、完整性、可用性和可信度的活动。入侵活动包括非授权用户试图存取数据、处理数据或者妨碍计算机系统正常运行的行为，以及授权用户滥用权力的行为。

所谓入侵检测(intrusion detection)，就是通过从计算机网络或计算机系统中的若干关键点收集信息并对其进行分析，从中发现网络或系统中是否有违反安全策略的行为和遭到袭击的迹象。

进行入侵检测的软件与硬件的组合就是入侵检测系统(Intrusion Detection System，IDS)。对于一个成功的入侵检测系统来讲，它不但可使系统管理员时刻了解网络系统(包括程序、文件和硬件设备等)的任何变更，还能给网络安全策略的制订提供指南。而且，它应该管理、配置简单，从而使非专业人员也非常容易地获得网络安全。另外，入侵检测的规模还应根据网络威胁、系统配置和安全需求的改变而改变。入侵检测系统在发现入侵后，会及时作出响应，包括切断网络连接、记录事件和报警等。

3.1.2 入侵检测系统的功能结构

一个典型的IDS从功能上可以分为3个组成部分：传感器(sensor)、分析器(analyzer)和管理器(manager)，如图3.1所示。

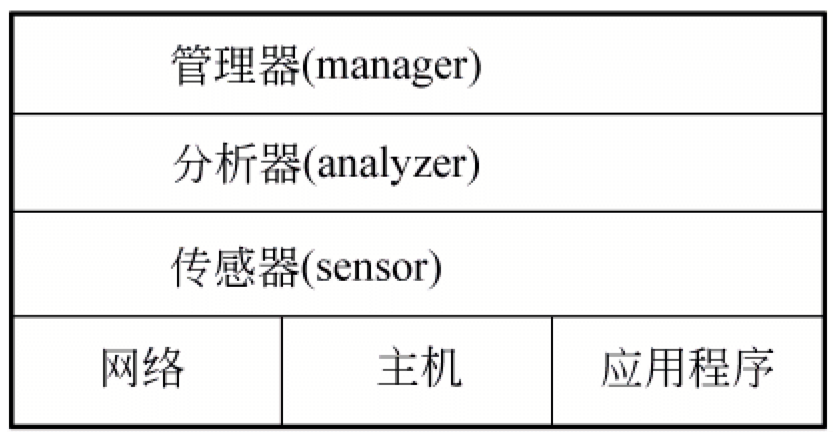


图3.1 IDS的功能结构

其中，传感器负责收集原始数据，包括任何可能包含入侵行为线索的数据，比如网络数据包、日志文件和系统调用的记录等。传感器将这些数据收集起来，然后发送到分析器进行处理。

分析器又称为检测引擎，它负责从一个或多个传感器处接收信息，并通过分析确定是否发生了非法入侵活动。分析器的输出为标识入侵行为是否发生的指示信号，例如一个警告信号。该指示信号中还可能包括相关的证据信息。另外，分析器还能够提供可能采取的对策的相关信息。

管理器通常也称为用户控制台，它以一种可视的方式向用户提供收集到的各种数据及相应的分析结果。用户可以通过管理器对IDS进行配置，设定各种系统参数，从而对入侵行为检测及相应对策进行管理。

另外，国际上一个致力于入侵检测系统标准化工作的组织(Common Intrusion Detection Framework，CIDF)阐述了一个入侵检测系统的通用模型，它将一个入侵检测系统分为以下组件：

(1) 事件生成器(Event Generators，简称为E-box)。

(2) 事件分析器(Event Analyzers，简称为A-box)。

(3) 事件数据库(Event Database，简称为D-box)。

(4) 响应单元(Response Unit，简称为R-box)。

CIDF将入侵检测系统需要分析的数据统称为事件(event)，它可以是基于网络的入侵检测系统中网络中的数据包，也可以是基于主机的入侵检测系统从系统日志等其他途径得到的信息。CIDF对于各部件之间的信息传递格式、通信方法和标准API进行了标准化。

按照CIDF的规定，上述4种IDS组件使用gidos(generalized intrusion detection objects)来交换数据，gidos在CIDF工作组的CISL(Common Intrusion Specification Language)文档中定义。一个gidos有以下几种作用：记录某个时间发生的某个事件；根据某些事件得出的汇总报告；提供一个执行某个动作的指令。

事件生成器从IDS监控的计算机环境中获得数据，并把它们转化为CIDF gidos格式。事件生成器处理的数据可以从审计记录中得来，也可以来自网络通信信息，或者来自SQL数据库中生成的事务信息。事件生成器在事件发生后尽量快地提供该事件的报告。事件分析器接收其他组件发送的gidos，对其进行分析并返回新的gidos(一般是事件的综合信息)。例如，事件分析器可以是一个统计比较工具，或者是一个特征检查工具，或者是一个事件汇总工具。事件数据库主要用于存储事件，它也可以存放其他各种中间的和最终的数据。响应单元接收其他单元递交的gidos并执行它们规定的动作，动作包括中止进程、连接复位和改变文件允许值等。

3.1.3　入侵检测系统的分类

根据分类的角度不同，入侵检测系统可以有不同的分类方法，大致有下面几种，如图 3.2 所示。

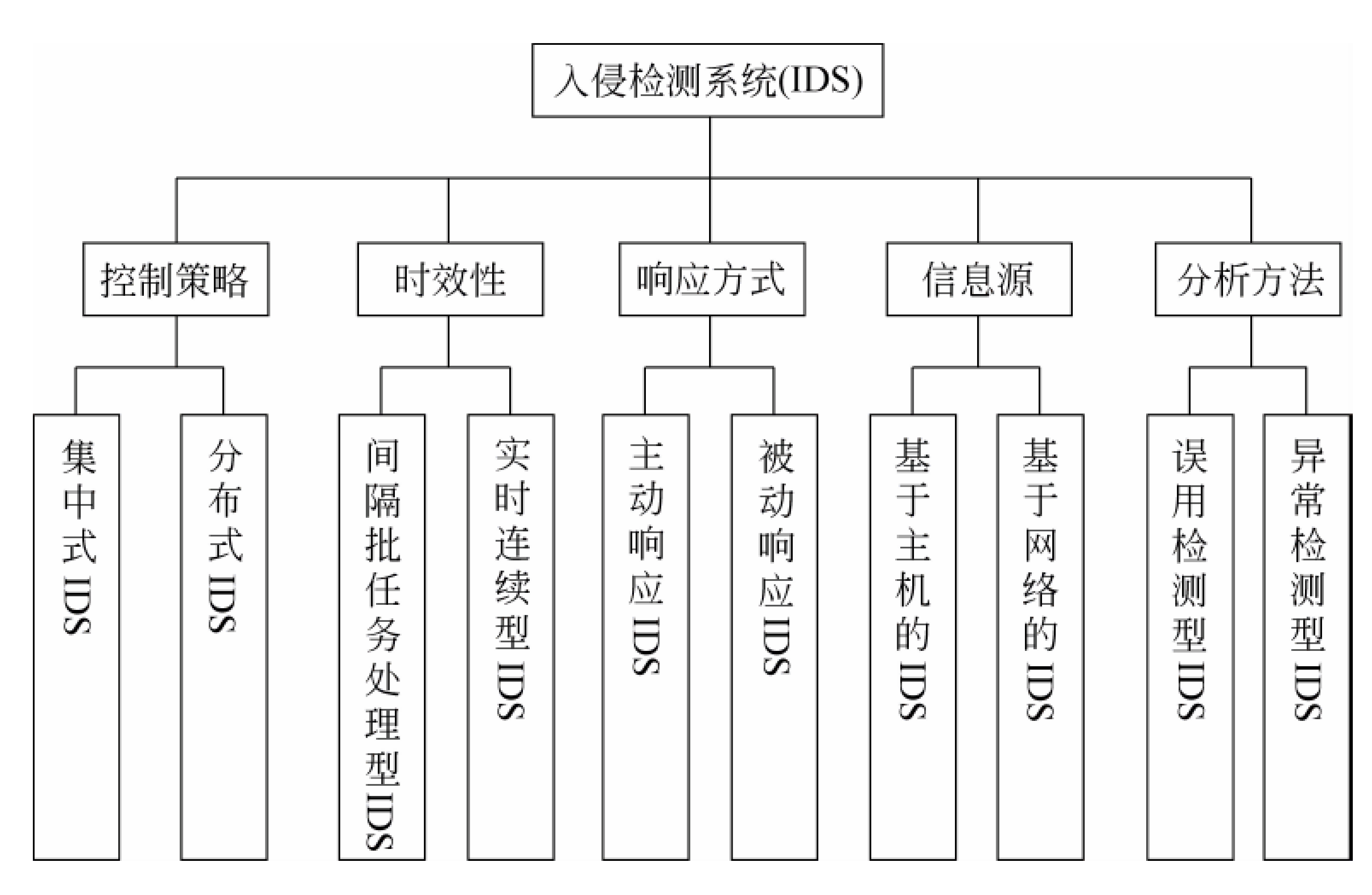


图 3.2　入侵检测系统分类

1. 按照分析信息的来源分类

这种分类方法主要依据分析数据收集的来源，将入侵检测系统分为以下两种。

1）基于网络的入侵检测系统

基于网络的入侵检测系统的数据源是网络上的数据包。一般来说，基于网络的入侵检测系统担负着保护整个网段的任务，需要获取和分析网络上传输的所有数据包。可以将某台主机的网卡设于混杂模式(promiscuous mode)，获取和分析网段内的所有数据包；或直接在路由或交换设备上放置入侵检测模块。

与基于主机的入侵检测系统相比，基于网络的入侵检测系统的最大优点是容易部署，不需要在受保护的机器上安全检测软件，不占用被保护设备的任何资源。另外，它还有隐蔽性好、检测速度快、视野宽广等优势。不过，它也有弱点，其一是它容易发出误警消息；其二是它容易受到网络流量大小的影响，当网络流量足够大时，可能出现丢包现象。

2）基于主机的入侵检测系统

基于主机的入侵检测系统往往以系统日志、应用程序日志等作为数据源，当然也可以通过收集主机的其他信息(如系统调用、登录尝试、文件访问与权限变化等)进行分析。基于主机的入侵检测系统保护的对象就是所在的主机。

与基于网络的入侵检测系统相比，基于主机的入侵检测系统的主要优势是它可以更精确地判断入侵事件，误警率比较低；并且不受网络流量大小的影响。不过，其缺点也显而易见，首先，它需要占用被保护主机的资源；其次，它不能检测到一些网络层的攻击。

2. 按照分析方法分类

这是比较传统的分类方法，把入侵检测模型大体上分为误用入侵检测模型(misuse detection model)和异常入侵检测模型(anomaly detection model)两种，如图 3.3 所示。

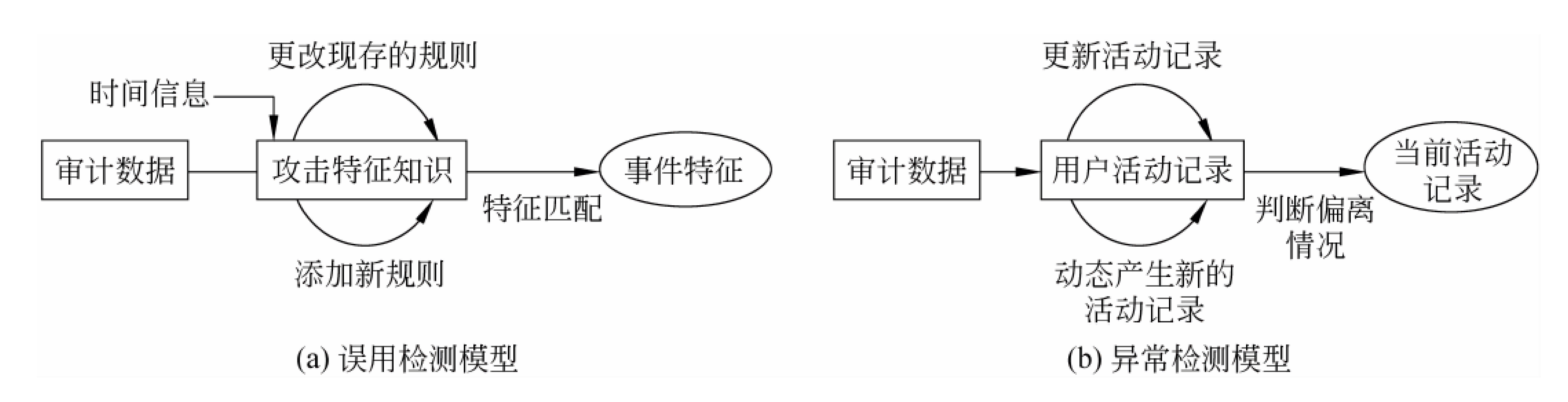


图 3.3 按照分析方法划分的入侵检测模型

3. 按照分析技术的时效性(工作方式)分类

依照分析技术的时效性(工作方式),可以把入侵检测系统分为以下两类。

(1) 实时入侵检测(在线入侵检测):实时联机的检测系统,它包含对实时网络数据包分析和对实时主机审计分析。

(2) 脱机入侵检测(离线入侵检测):在事后分析审计事件,从中检查入侵活动,是一种非实时工作的系统。

4. 按照系统的控制策略分类

按照系统的控制策略和各个模块运行的分布方式不同,可以将入侵检测系统分为以下两种。

(1) 集中式入侵检测系统:检测系统的各个模块(数据收集、分析和响应)的运行都集中在一台计算机上运行,适合比较简单的网络环境。

(2) 分布式入侵检测系统:检测系统的各个模块分布在多台计算机或设备上运行,可以有多个数据采集器和分析器,一般使用层次结构组织。

5. 按照响应方式分类

按照检测到报警后系统的响应方式不同,可以将入侵检测系统分为以下两种。

(1) 主动响应入侵检测系统:检测到报警后自动采取响应行为,包括收集辅助信息,改变环境以堵住导致入侵发生的漏洞,对攻击者采取行动等。

(2) 被动响应入侵检测系统:采用报警、通知、报告和存档等方式将信息提供给系统用户,依靠管理员在这一信息的基础上采取进一步的行动。

3.1.4 入侵检测的分析方法

如前所述,根据检测和分析方法的不同,入侵检测技术主要分为两大类:基于特征(signature-based)的入侵检测(又称误用检测,misuse detection)和基于异常(anomaly-based)的入侵检测。

1. 基于异常的入侵检测

基于异常的入侵检测根据用户的异常行为或者对资源的异常存取来判断是否发生入侵事件。例如,一个用户 A 一般在早上 9 点到晚上 5 点之间登录到服务器,如果有一天,服务器发现该用户账号在午夜 12 点登录到服务器来,就认为是一次入侵事件。基于异常的入侵检测要建立一个阈值来区分正常事件与入侵事件。

基于异常的入侵检测因为与具体系统无关，通用性较强，甚至有可能检测出以前未出现过的攻击方法。不过，因为难以对整个系统内的所有用户行为进行全面的描述，况且每个用户的行为是经常改变的，所以它的主要缺陷在于误检率较高。实际上，基于异常的入侵检测方法的最大不足之处是“异常并不等于入侵”，如图 3.4 所示。

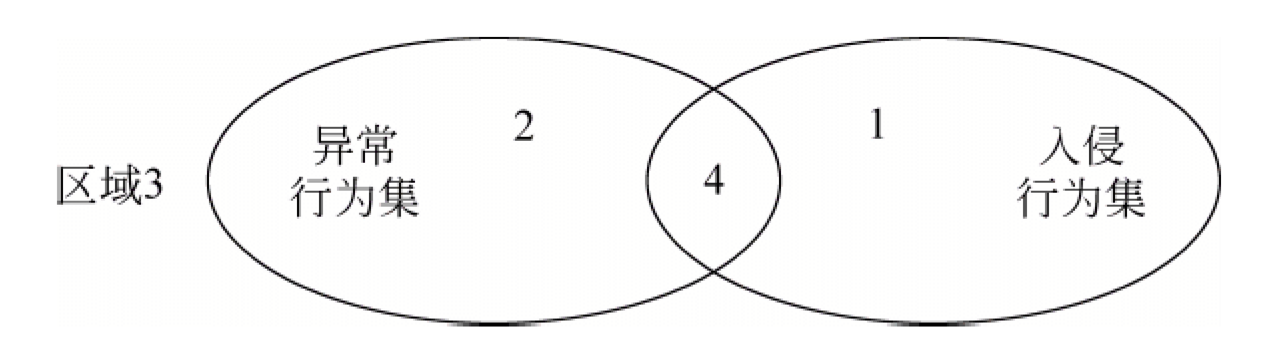


图 3.4　异常与入侵的比较

根据图 3.4，可以把异常与入侵的关系分为下面 4 种类型：

(1) 对应图 3.4 区域 1 的部分，是入侵但不是异常。虽然它是一个入侵事件，但由于不在异常集中，所以系统不能检测。这类事件的集合称为错误拒绝集(false negatives)。

(2) 对应于图 3.4 区域 2 的部分，不是入侵但是异常。虽然不是一个入侵事件，但是由于在异常集中，所以会被认为是一次入侵事件。这类事件的集合称为错误接受集(false positives)。

(3) 对应于图 3.4 区域 3 的部分，不是入侵也不是异常。这类事件会被正确排除，称为正确拒绝集(true negatives)。

(4) 对应于图 3.4 区域 4 的部分，是入侵同时也是异常。这类事件会被正确检测，称为正确接受集(true positives)。

基于异常的入侵检测用于检测异常活动的发生，如果有异常活动发生就认为是有入侵嫌疑。既然有异常的概念，就有相应的正常活动的定义，并且要对正常活动的偏离定义阈值。基于异常的入侵检测方法主要有以下几种。

1) 统计方法(Statistical Approach)

基于异常的入侵检测中经常使用统计方法，假设正常的操作存在内在的统计规律。入侵检测系统观测主体(如用户)的活动并生成相应的活动特征表(profile)。活动特征表含有若干指标(measure)值，每个指标值代表系统安全性某个方面的阈值。这些指标值根据经验值或一段时间内的统计得到。假设 $S_1, S_2, \cdots, S_n$ 代表指标 $M_1, M_2, \cdots, M_n$ 的阈值，然后从某段时间的审计记录中提取当前特征表($T_1, T_2, \cdots, T_n$)，如果 T_i 比 S_i 大，则表示 i 指标的异常度大。另外也可以使用各指标阈值的带权平方和来做比较标准，带权平方和的表达式如下：

$$A_1 S_1^2 + A_2 S_2^2 + \cdots + A_n S_n^2$$

其中，A_i 为 S_i 的权重，$A_i > 0$。

特征表一般含有以下类型的指标。

(1) 活动强度指标：用来衡量用户活动的频率，如一分钟内用户产生的审计记录数。

(2) 审计记录-分布情况指标：用来衡量最近审计记录中所有活动类型的分布情况，如某个用户的文件存取与 I/O 活动的分布情况。

(3) 类别指标：用来在不同类别中衡量某一活动的分布情况，如来自不同物理位置的 login 命令的相对频率，或系统中每个邮件服务器、编译器、Shell 和编辑器的相对使用频率。

(4) 序数指标：用数值来表示活动情况，如某一用户对 CPU 和 I/O 的使用总值。

使用统计方法的优点是可以利用相对成熟的统计理论成果。但使用统计方法的一个主要缺点是入侵者可以“训练”该 IDS 使得它把异常逐渐当作正常；另一个缺点是难以确定合适的阈值。另外，有些行为难以用纯统计的方法来建模。

2）神经网络（neural network）

利用神经网络检测入侵的基本思想是用一系列信息单元（命令）训练神经单元，这样在给定一组输入后，就可能测出输出。具体来说，它首先搜集一些命令集对神经网络进行训练，然后输入当前命令和前 W 个命令（W 为已执行命令窗口的大小），输出是预测的下一个命令。

与统计方法相比，神经网络更好地表达了变量间的非线性关系，并且能自动学习并更新。它能更好地处理原始数据的随机特性，即不需要对这些数据作任何统计假设，并且有较好的抗干扰能力。这种方法的最大缺点是不能检测神经网络输入集以外的入侵事件，因为它不满足神经网络的输入事件。命令窗口 W 的大小也难以选取，窗口太小，则网络输出不好；窗口太大，则网络会因为大量无关数据而降低效率。

2. 基于特征的入侵检测

基于特征的入侵检测又称误用检测（misuse detection），它事先把入侵者活动用特征模式表示，在检测时将已有的攻击特征与用户的活动进行比较，并以此判断是否发生了攻击行为。例如，著名的 Internet 蠕虫事件就是利用了 fingerd 和 sendmail 的漏洞进行攻击，对这种攻击就可以使用这种检测方法。

基于特征的入侵检测由于依据具体的特征库进行判断，所以检测准确度很高，并且因为检测结果有明确的参照，也为系统管理员做出相应的措施提供了方便。使用基于特征的入侵检测的一个不足之处是这种方法需要对攻击进行编码，问题是并不是所有的攻击都可以用编码的方式来很好地表达，而且这种方式不能检测未知的攻击方式，尤其难以检测出内部人员滥用权力和泄露机密的行为。基于特征的入侵检测方法大致有以下 3 种。

1）特征字符串匹配

这是一种最简单的入侵检测方式，也是很有效的检测方式。对于很多种入侵手段，如 FTP 攻击、远程缓冲区溢出攻击、CGI 攻击等，都会包含一些特征字符串。例如，FTP wuftp260 Siteexec 攻击包含特征字符串"SITE EXEC %p"，IMAP-x86-linux-buffer-overflow 攻击包含特征字符串"|E8C0 FFFF FF|/bin/sh"，WEB-CGI-PHP CGI 攻击包含特征字符串"php.cgi?/"，等等。通过特征字符串匹配，可以立即发现正在进行的入侵。特征字符串匹配实现简单、快速，检测结果明确，便于做出响应。

这种方法的缺点是，入侵者可以通过一系列的方法来躲过这种入侵检测。例如，通过网络上数据打包的不同方式，利用不同操作系统对网络数据包的不同处理，采用不同的指令，等等，来隐藏特征字符串，从而躲过检测。

2）专家系统

专家系统在已有知识的基础上做出入侵判决。知识基于已知的入侵行为、已知的系统漏洞、安全策略的配置失误以及期望的系统行为。但是，对于知识库中未知的入侵行为，这种方式无法进行入侵检测。

专家系统是基于特征的检测中运用得最多的一种方法。采用专家系统对入侵进行检测，经常是将入侵的特征知识转化为 IF-THEN 结构的规则。IF 部分为入侵特征，THEN 部分为系统防范措施。当 IF 部分的条件全部满足时，触发 THEN 部分的防范措施。其中，

IF-THEN 结构构成了具体攻击的规则库，状态行为及其语义环境可根据审计事件得到，推理机根据规则和行为完成判断工作。

在具体实现中，专家系统面临的主要问题是难以保证知识库的完备性，并且处理效率比较低。因为这些缺陷，专家系统技术目前只是在各种研究原型中得到应用，而商业化的产品采用了其他效率更高的技术，其中目前应用最广泛的就是特征分析技术。与专家系统一样，特征分析也需要知道攻击行为的具体知识。但是，攻击方法的语义描述不是被转化为检测规则，而是在审计记录中能直接找到的信息形式。这样，就无须像专家系统一样需要处理大量数据，从而大大提高了效率。

3）状态转换分析

状态转换分析技术是 UC Santa Barbara 大学研发的入侵检测技术。入侵攻击被表示为被监测系统上的一系列状态转换。被监测系统可能处于不同的状态下，状态和状态之间的转换需要由一些关键的行动来触发。当一系列事件被识别为一个关键行动后，系统就从一个状态转换成下一个状态。系统最初的状态是安全、正常的状态，最后的状态是发现入侵的状态。不同的入侵手段需要用不同的状态转换规则去描述。

状态转换分析技术识别的是系统的状态和关键的行动，攻击者很难利用掺杂一些别的命令来对实施攻击的命令序列进行隐藏。只要攻击会使系统产生那些必经的状态，状态转换分析技术就不需要对具体的入侵步骤和方法有明确的了解，因此在一定程度上状态转换分析技术可以检测未知的入侵。

3.2 基于行为检测的信任度评估模型

3.2.1 模型框架

本节所述的基于网络行为检测的信任度评估模型（以下简称为 NBTVE 模型）包括 3 个功能层，每层又包含多个功能模块，如图 3.5 所示。3 个功能层分别是网络行为信息采集层、网络行为分析层和信任度评估层。其中网络行为信息采集层包括数据包采集、协议解析和网络行为还原 3 个功能模块；网络行为分析层包含模式匹配和统计分析 2 个功能模块；信任度评估层包括信任度初始化、信任度积分提升/降低和信任度计算及更新 4 个功能模块。

网络行为信息采集层主要是应用网络数据包嗅探技术分析用户的网络行为。其中数据包采集模块主要是应用网络监听技术抓取数据包。协议解析模块是实时对分组内容进行匹配，根据数据包的类型和所处的网络层次对数据包进行协议解析。网络行为还原模块主要是对采集的数据包进行重组，分析用户的行为。

网络行为分析层主要是对采集到的行为信息进行判断和识别，通过模式匹配和统计分析判断用户行为是正常行为还是恶意行为，如果是恶意行为，给出报警级别，并将这些信息记录到知识库当中。知识库中记录了用户的历史行为，并对用户的网络行为进行了分类。通过这种方式可以为每个用户建立一个行为档案，并分析用户正常行为的属性，及时地更新规则库。

信任度评估层是 NBTVE 的重点，信任度评估层对采集的数据进行分析，根据本章设计的信任度评估算法，对每个用户的信任度作出评价。信任度评估层主要涉及以下问题：(1)初始信任度；(2)信任度积分的提升和降低；(3)信任等级的计算。

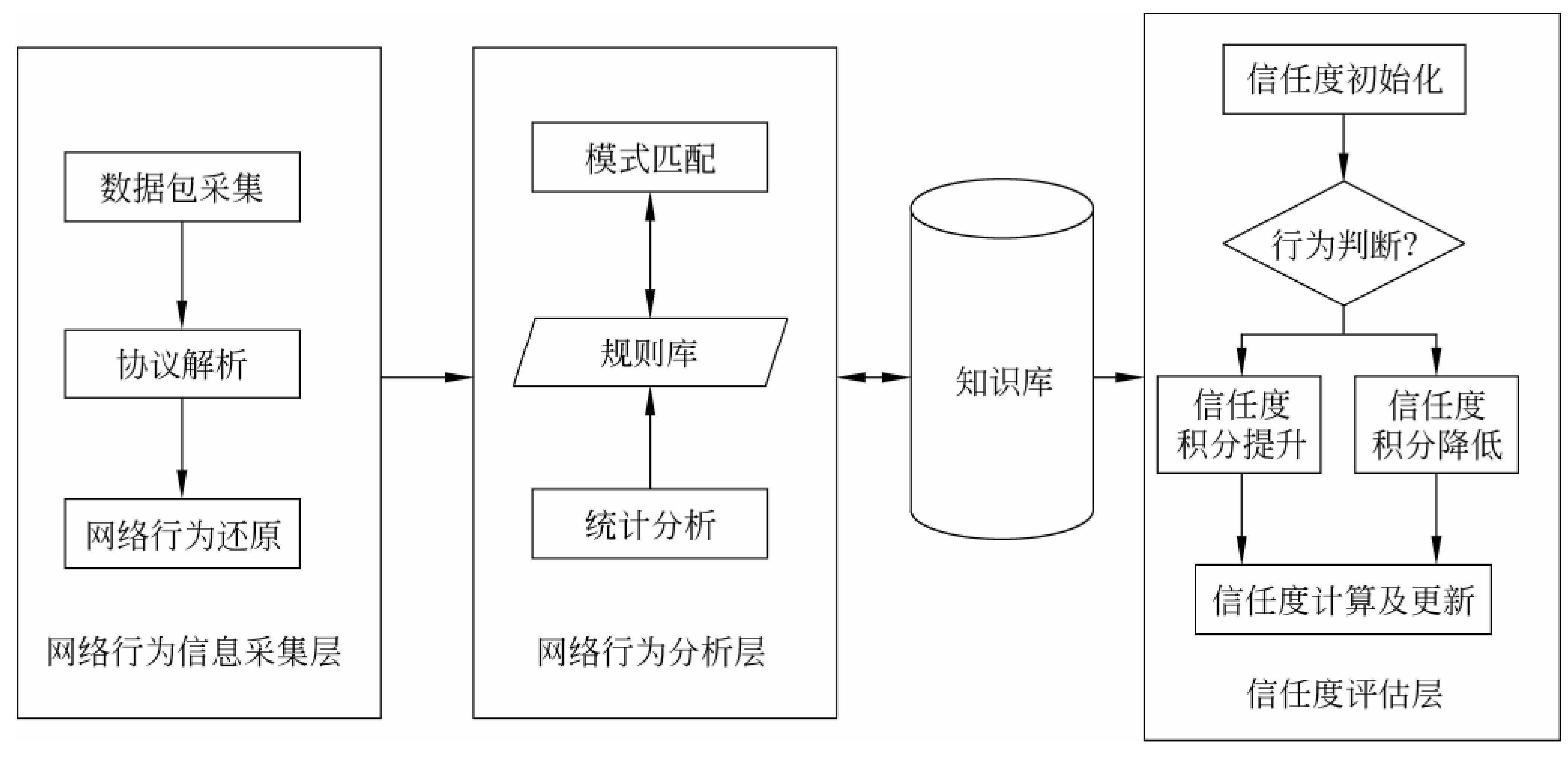


图 3.5 NBTVE 模型功能模块

3.2.2 工作流程

图 3.6 给出了 NBTVE 模型的工作流程，网络行为信息采集层首先应用入侵检测系统技术采集数据包，然后进行协议解析并对数据包进行重组，还原用户的行为。这些信息将交给网络行为分析层，进行用户行为模式匹配，并上传到知识库。信任度评估层结合用户网络

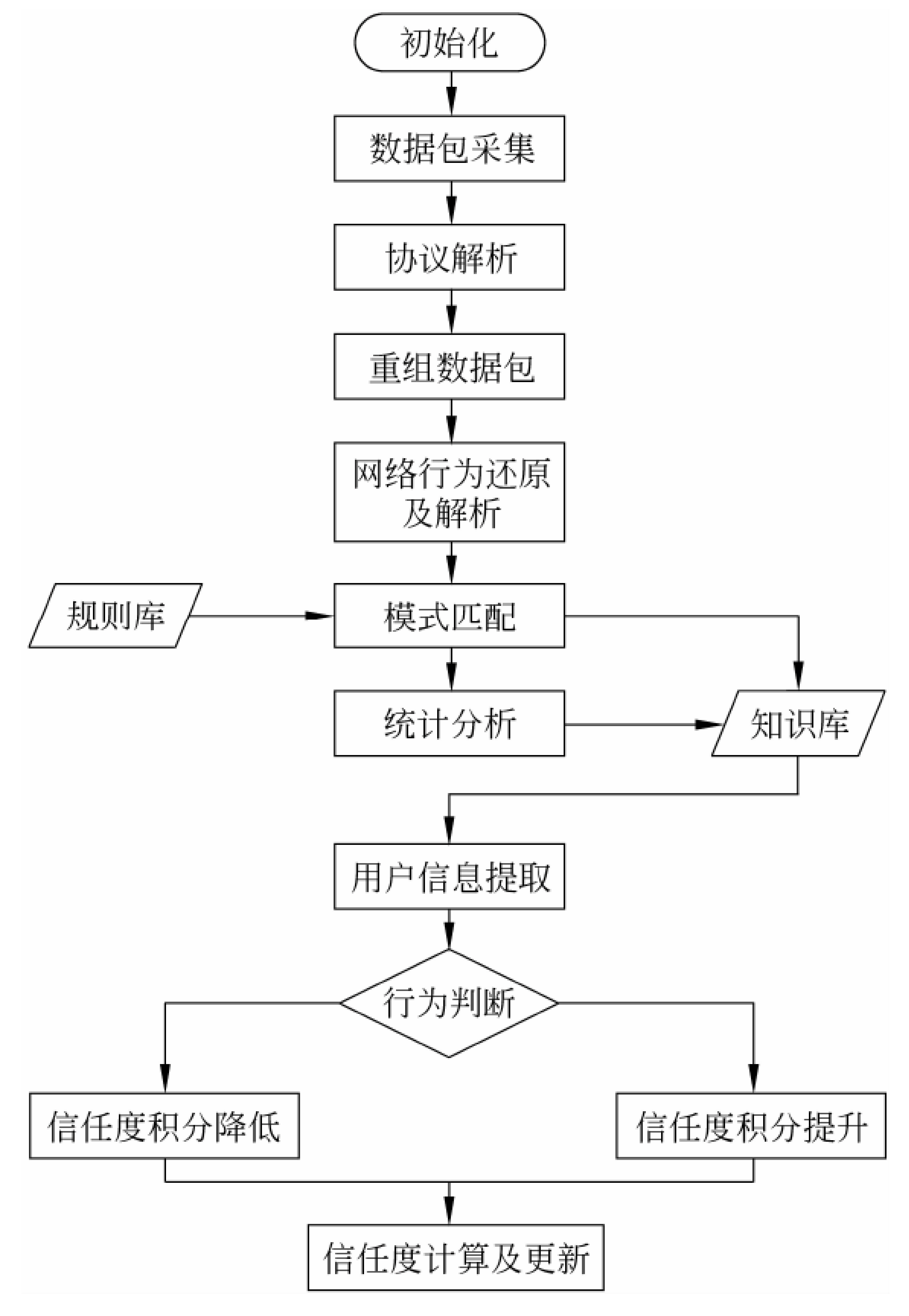


图 3.6 NBTVE 模型的工作流程

行为对用户的信任度和信任等级作出综合的评价。

NBTVE 模型中的规则库是记录不正常行为的数据库，统计分析模块通过学习用户行为，对用户行为的正常属性值进行记录，如果用户的行为超出正常值范围则认为是不正常行为。规则库中也包括一些基本的非法行为的信息，如特洛伊木马和 ping 扫描攻击等。

NBTVE 模型中的知识库是记录用户网络行为的数据库，网络行为分析层采集到的信息通过分类整理提交到知识库。直接信任度计算时所应用到的数据是从知识库中查询得到的。知识库是连接网络行为分析层和信任度评估层的桥梁。

3.3　基于行为检测的信任度评估算法

3.3.1　信任度表示与度量

1. *初始信任度*

日常生活中，实体之间的信任关系是通过实体交互建立起来的。在信任度评估中，实体之间的信任关系也是通过用户的日常表现给出对用户的一个评价。对于一个陌生的实体，算法将它的初始信任度积分设置为 0，初始信任等级设置为 0。通过每次对实体的网络行为以及交易行为情况进行评估，系统会不断调整用户的信任度。

2. *信任度的表示*

本算法中信任度的表示是采用(Tc，Tg)这对参数来表示的，Tc 代表用户的信任等级，Tg 代表用户的信任度积分。首先，随着用户正常行为的增加，用户的信任度积分会不断地提高，当用户的信任度积分提高到一定的程度，用户的信任等级也会相应地改变。用户的信任等级与信任度积分的换算过程中一个重要的参数是信任等级所需最低积分 MinGi，这一参数设定各个等级的信任度积分的最低阈值。Tg 与 Tc 的关系可以表示如下：

$$\mathrm{Tc} = \mathrm{Max}\ i\ (\mathrm{Tg} >= \mathrm{MinGi}) \tag{3.1}$$

在本算法中，用户的信任度被量化为 7 个信任等级，如表 3.1 所示。

表 3.1　用户信任等级

信任等级	说　　明	信任等级	说　　明
−1	不可信用户信任等级	3	用户信任度较好
0	用户初始信任等级	4	用户信任度很好
1	用户信任度较低	5	用户信任度非常高
2	用户信任度一般		

3.3.2　算法描述

首先通过每次对实体的网络行为及交易行为情况进行评估，系统会不断调整用户的信任度。随着用户网络行为或者交易行为次数的增加，用户的信任度应该随之改变，用户的信任等级也将随之改变。算法针对两种类别的用户行为进行不同的处理。对于网络行为，若用户本次行为被检测为违规行为，则根据用户的信任度等级、用户的具体违规行为及警报级别等信息计算用户将要被减少的信任度积分，然后再将原有信任度积分减去这个值；反之，

如果用户的行为被检测为正常行为，则对信任度积分进行提升计算。对于交易行为，若用户的交易成功，那么算法会根据用户的信任等级对信任度进行提升；用户交易失败，算法会根据用户的信任等级对信任度进行相应的降低。

具体算法如下：

当一个用户的行为被检测为违规行为或者失败的交易行为，用户的信任度积分就会随之改变：

$$\Delta T_g = B_p \times U_p \times D_p \times G_p \tag{3.2}$$

$$T_g = T_g - \Delta T_g \tag{3.3}$$

其中，信任度惩罚基数 B_p：对用户信任度积分的调整是在此参数的基础上进行的。这是一个调整用户信任度的基本参数，相当于调整信任度的一个基本单位。

信任度惩罚因子 U_p：这是对信任度进行调整的一个重要参数，它设置的目的在于对信任等级不同的用户的违规行为处以不同程度的惩罚。用户的信任等级越高，算法中对用户的惩罚就会越严厉。

信任度惩罚力度 D_p：这是对信任度进行调整的另一个重要参数，这个参数是根据用户的具体行为来进行设置的。根据对用户行为的分析，将此参数设置为 6 个类别，分别对应表 3.2 中的 6 种级别，即不同的违规级别有不同的惩罚力度。这个参数的设定需要对用户的行为进行一定的分析和研究，进行统计分类。表 3.2 给出了参数的取值。其中的违规行为级别参考文献[3]中给出的事件安全级别。

表 3.2　违规行为级别[3]

违规级别	事件描述	代表事件	惩罚力度 D_p
1 级	探测主机是否存活的事件	ping 事件	1
2 级	收集操作系统或开放服务信息的事件	tcp 扫描事件	2
3 级	密码探测失败事件	ftp 密码探测事件	4
4 级	远程漏洞攻击失败事件	Unicode 漏洞测试失败	8
5 级	远程漏洞攻击成功事件	Windows 系统漏洞攻击成功	16
6 级	系统被攻陷	非法用户盗取 root 权限	32

行为类别惩罚因子 G_p：这是对信任度进行调整的一个重要参数，由于用户的网络行为与交易行为对信任度的影响程度不同，所以用户不同类别的行为对应不同的 G_p。此参数控制网络行为与交易行为的权重。

式(3.2)是本次信任度积分的负增量。B_p 是一个基数，U_p、G_p 和 D_p 分别根据用户的信任等级、用户的行为类别和用户的违规行为级别设置。在 B_p 的基础上计算，在原信任度积分的基础上减少 ΔT_g，得到新的信任度积分。

同理，若用户本次行为被检测为正常行为或成功的交易行为，则根据用户的信任等级、用户的违规行为及总行为次数等信息计算用户将要增加的信任积分。

$$\Delta T_g = B_a \times U_a \times G_a \tag{3.4}$$

$$T_g = T_g + \Delta T_g \tag{3.5}$$

其中，信任度提升基数 B_a：对用户信任度积分的调整是在此参数的基础上进行的。这是一个调整用户信任度的基本参数，相当于调整信任度的一个基本单位。此参数与信任度惩罚

基数对应，用来区别用户的正常行为和违规行为。

信任度奖励因子 U_a：这是对信任度进行调整的一个重要参数，设置它的目的在于对信任等级不同的用户的正常行为给予不同程度的奖励。用户的信任等级越高，算法中对用户的奖励就会越吝啬。

信任度提升因子变量 G_a：这个参数是针对用户行为的类别设计的。由于用户的网络行为与交易行为对信任度的影响程度不同，所以用户不同类别的行为对应不同的 G_a。此参数控制网络行为与交易行为的权重。

式(3.4)中是本次信任度积分的增量。B_a 是一个基数，U_a 和 G_a 分别根据用户的信任等级和用户的行为类别设置。在 B_a 基础上计算，在原信任度积分的基础上增加 ΔT_g，得到新的信任度积分。

除了考虑用户网络行为及交易行为对其信任度积分的影响，还要考虑到时间因素。用户长时间没有登录网络，则不能完全根据用户以前的行为来判断用户的可信度，用户的可信度降低。

时间衰减因素：每隔固定时间进行用户信任度的衰减计算。时间因素对用户的信任度有一定影响，但不是主导因素。这个衰减应该是呈现边际递增的，这条曲线满足以下两个条件：

当 $T_n - T_l \to 0$ 时：

$$\lambda(T_n, T_l) = 1$$

当 $T_n - T_l \to \infty$ 时：

$$\lambda(T_n, T_l) = 0$$

T_l 表示用户最后的上网时间，这个参数用于记录用户最后一次的上网行为发生的时间。T_n 表示现在的时间。这里 T_l 和 T_n 的单位是天。本章将信任度衰减函数设计为

$$\lambda(T_n - T_l) = e^{(-T_n - T_l)/C} \tag{3.6}$$

$$T_g = T_g \times \lambda(T_n - T_l) \tag{3.7}$$

C 是算法中的重要参数，它影响着节点信任度变化的速度。可以调整常数 C 的大小来控制时间衰减因素对用户信任度的影响程度。

3.3.3　算法实例

假设用户 A 现在的信任度积分是 5，A 被检测到发生一次网络行为，若此网络行为为正常行为，那么应该执行信任度提升算法。首先根据环境以及前述算法中各参数的说明设置参数，如表 3.3 和表 3.4 所示。

表 3.3　信任度参数

信任等级(T_c)	所需最低积分(MinGi)	奖励因子(U_a)	惩罚因子(U_p)
−1	$-\infty$	1	0.325
0	0	1	0.325
1	10	0.95	0.35
2	30	0.85	0.4
3	60	0.7	0.5
4	100	0.5	0.7
5	150	0.25	1

表 3.4　常用参数

变　　量	数　　值	说　　明
信任度惩罚因子(G_p)	0.2/0.4	G_p 和 G_a 的两个值分别对应网络行为和交易行为
信任度奖励因子(G_a)	0.2/0.4	
信任度惩罚基数(B_p)	0.6	
信任度提升基数(B_a)	0.2	

对用户 A 首先执行衰减算法，由于衰减算法主要是针对长期没有操作的用户而做的衰减计算，假设本次执行不满足衰减条件，则可执行信任度提升算法(式(3.4)和式(3.5))可得：

$$\Delta T_g = 0.2 \times 1 \times 0.2 = 0.04$$
$$T_g = 5 + 0.04 = 5.04$$

若检测到 A 有 ftp 密码探测行为，且此行为是违规行为，违规级别为 3，则执行信任度衰减算法(式(3.2)和式(3.3))可得：

$$\Delta T_g = 0.6 \times 0.325 \times 4 \times 0.2 = 0.156$$
$$T_g = 5 - 0.156 = 4.844$$

3.3.4　实验分析

为了测试不同参数对信任度积分变化的影响，本节设计了实验 1 和实验 2。在实验 1 中，假设用户 A、B、C、D 的初始信任度积分均为 59.5。用户 A、B、C、D 分别发生一次级别为 1、2、3、4 的违规行为。在实验 2 中，假设用户 A、B、C、D 的初始信任度积分均为 60。用户 A、B、C、D 分别发生一次级别为 1、2、3、4 的违规行为。根据 NBTVE 算法可得到各用户的信任度积分，实验结果如图 3.7 所示。从这两组数据的对比中可以看出，违规级别越严重，用户受到的惩罚就越大。对用户的惩罚也呈边际递增的规律。从图 3.7 中还可以看到，用户的信任等级越高，惩罚力度越大，即用户的信任度越高，对用户的要求越严格。

为了检测 NBTVE 算法的有效性，在实验 3 中我们将 NBTVE 算法与基于交易行为的信任度评估模型[4]中的算法进行了对比，以下简称算法 1 和算法 2。在本实验中设置前提条件如下：用户 A 的初始信任度为 0.6，在此基础上用户 B 与用户 A 发生 4 次成功的交易，并且在第二次交易发生后用户 A 被检测到对某 ftp 服务器进行密码测探。根据算法 1(采用表 3.3 和表 3.4 中的参数)与算法 2 分别计算用户 A 从初始信任度为 0.6 到每次交易后的信任度变化情况。算法 1 中的信任度积分等价于算法 2 中的信任度。实验结果如图 3.8 所示，M1 即算法 1，M2 即算法 2。算法 1 考虑了用户的网络行为，因此当用户 A 被检测到违规行为时对其进行了惩罚，A 的信任度降低。用户 A 的行为是具有极强的危害性的，NBTVE 算法中综合考虑用户的日常网络行为可以避免对用户的认识不全面。用户虽然在交易中表现正常，但存在网络入侵行为仍然表明用户具有潜在的危险性。将用户网络行为作为建立信任关系的参考依据可以充分地认识用户，使信任度评估结果更加真实可靠。

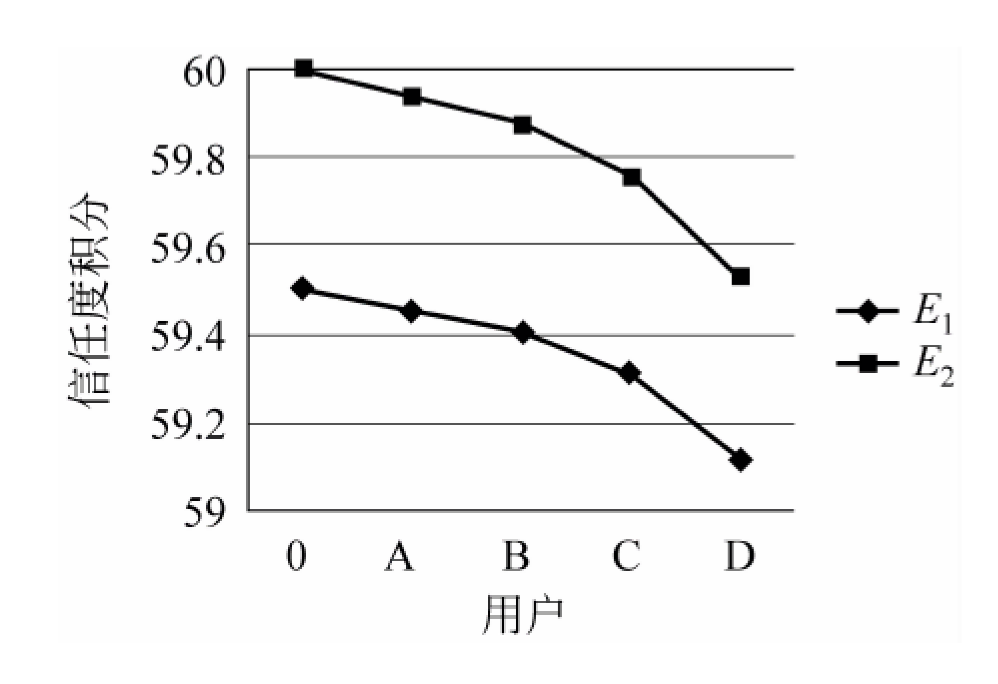


图 3.7 不同参数与信任度积分的关系

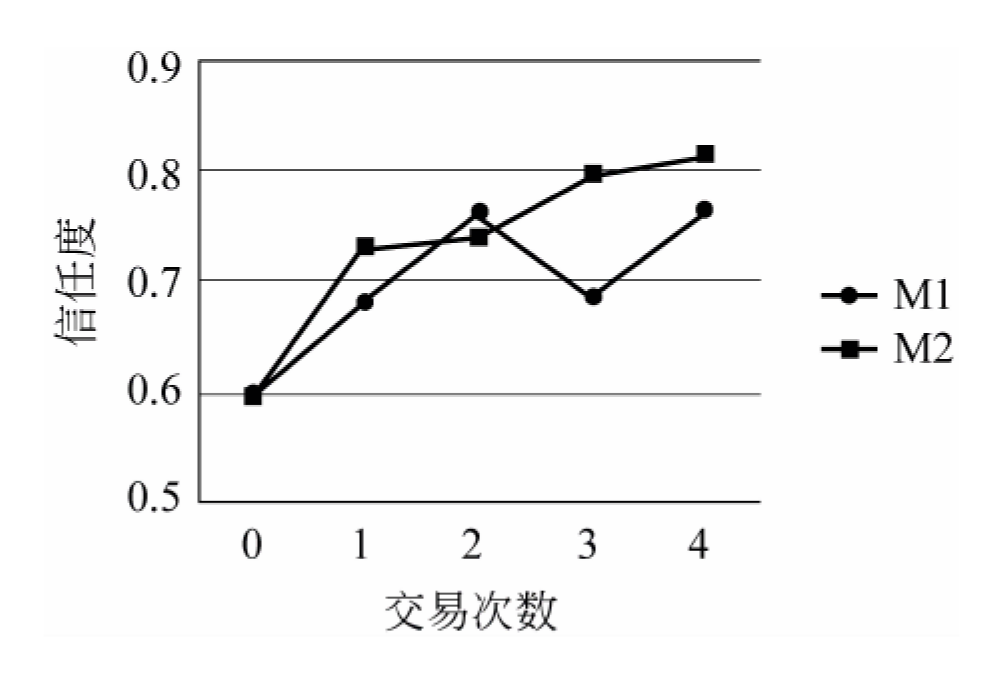


图 3.8 NBTVE 算法与基于交易行为的信任度评估算法对比

3.4 本章小结

本章专门阐述了一种基于网络操作行为的信任度评估模型和算法。首先从网络行为检测技术出发，详细介绍了入侵检测系统的功能、分类及分析方法；其次在此基础上提出了一种基于行为检测的信任度评估模型（NBTVE），该模型着重考虑了用户的网络操作行为，以用户的日常网络操作行为和交易行为作为信任度评估的依据。本章详细讨论了该信任度评估模型的框架、工作流程和评估算法，并通过实验分析了 NBTVE 模型及算法的有效性。

参 考 文 献

[1] 曹天杰，张永平. 信任管理在电子商务中的实现[J]. 计算机应用与软件，2003，28(3)：15-17.

[2] 王惠芳. 开放分布式系统中的信任管理[J]. 计算机工程，2004，28(8)：117-119.

[3] Xu HongXing，Shan Zheng，Lu Weifeng. Research and Implementation of a Level-based Network Intrusion Detection System. Computer Engineering，2002，28(10).

[4] 张仕斌，何大可，盛志伟. 信任管理模型的研究与进展[J]. 计算机应用研究，2006，7：18-22.

[5] 唐文，陈钟. 基于模糊集合理论的主观信任管理模型研究[J]. 软件学报，2003，14 (8)：1401-1408.

[6] 杨东浩，蒋文保. 分布式环境下的管理模型研究[J]. 北京机械工业学院学报，2008，23(2)：53-57.

[7] Jiang Wenbao，Guo Shaoxu，Chen Wenliang. A Trust Evaluation Model and Algorithm Based on Network Behavior Detection. In：2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology，October 2010.

第 4 章 自适应自动信任协商模型

第 1 章介绍了自动信任协商(ATN)的概念,即“通过凭证、访问控制策略的交互披露,资源的请求方和提供方自动地建立信任关系”。目前,虽然国内外自动信任协商研究发展迅速,但由于研究历史很短,还存在许多不足。例如,现有的大多数 ATN 系统中访问控制策略和协商策略基本上都是静态的,不能根据环境的变化而动态调整策略,不能有效兼顾效率和安全两方面的需求。有些系统简单地采用热心策略(eager strategy),即在信任协商前,请求者接收到协商方的访问控制策略后,将所可能满足策略的证书全部提交。而有些系统则采用吝啬策略(parsimonious strategy),即资源方要求什么证书,访问者才提交什么证书,通过迭代交互来建立信任关系。热心策略中证书交换次数相对较少,协商的效率高,但是协商过程中暴露了一些不必要的证书,可能导致敏感信息的暴露;吝啬策略极大程度地保护了用户的信息,但是信任协商过程复杂,协商效率不高。因此,目前尚缺乏一种既高效又能保证安全级别的协商机制。

为了解决以上问题,我们提出了一种新的信任协商模型,即一种自适应自动信任协商(Adaptive Automated Trust Negotiation,AATN)模型,它能根据信任度评估结果动态调整访问控制策略和协商策略,以有效兼顾信任协商中效率和安全两方面的需求。本章 4.1 节阐述 AATN 模型的框架结构,4.2 节介绍 AATN 的工作流程,4.3 节重点论述 AATN 中采用的自适应策略模式,4.4 节分析 AATN 的一致性校验器及校验算法。

4.1 自适应自动信任协商模型框架

AATN 模型提出了一种自适应信任协商策略,该策略不同于传统的吝啬策略或热心策略,而是根据用户信任度动态调整访问控制策略和证书的披露规则,对具有不同信任度的用户使用不同的访问控制策略,即对于信任度较高的用户使用相对宽松的访问控制策略,对信任度较低的用户使用相对严格的访问控制策略。自适应自动信任协商策略的特点在于:

(1) 在信任度高的用户之间相互协商时,通过选择宽松的访问控制策略来减少敏感证书的披露,从而简化协商过程,大大提高信任协商的效率。

(2) 在信任度低的用户之间相互协商时,通过选择严格的访问控制策略来严格审查协商的双方,从而使得协商过程变得更为谨慎,大大提高信任协商的安全性。

通过采用自适应自动信任协商策略,AATN 模型能够灵活动态地建立信任关系,可以根据信任度评估结果动态调整访问控制策略和协商策略,以有效兼顾自动信任协商中效率和安全两方面的需求。自适应自动信任协商区别对待每个请求者,对于相同的服务或者资源,协商策略根据用户的信任度,要求不同的请求者披露不同的凭证集,进而建立信任关系。

如图 4.1 所示,自适应自动信任协商模型主要包括信任凭证库、访问控制策略库、协商策略、信任度评估、一致性校验器以及协商协议。

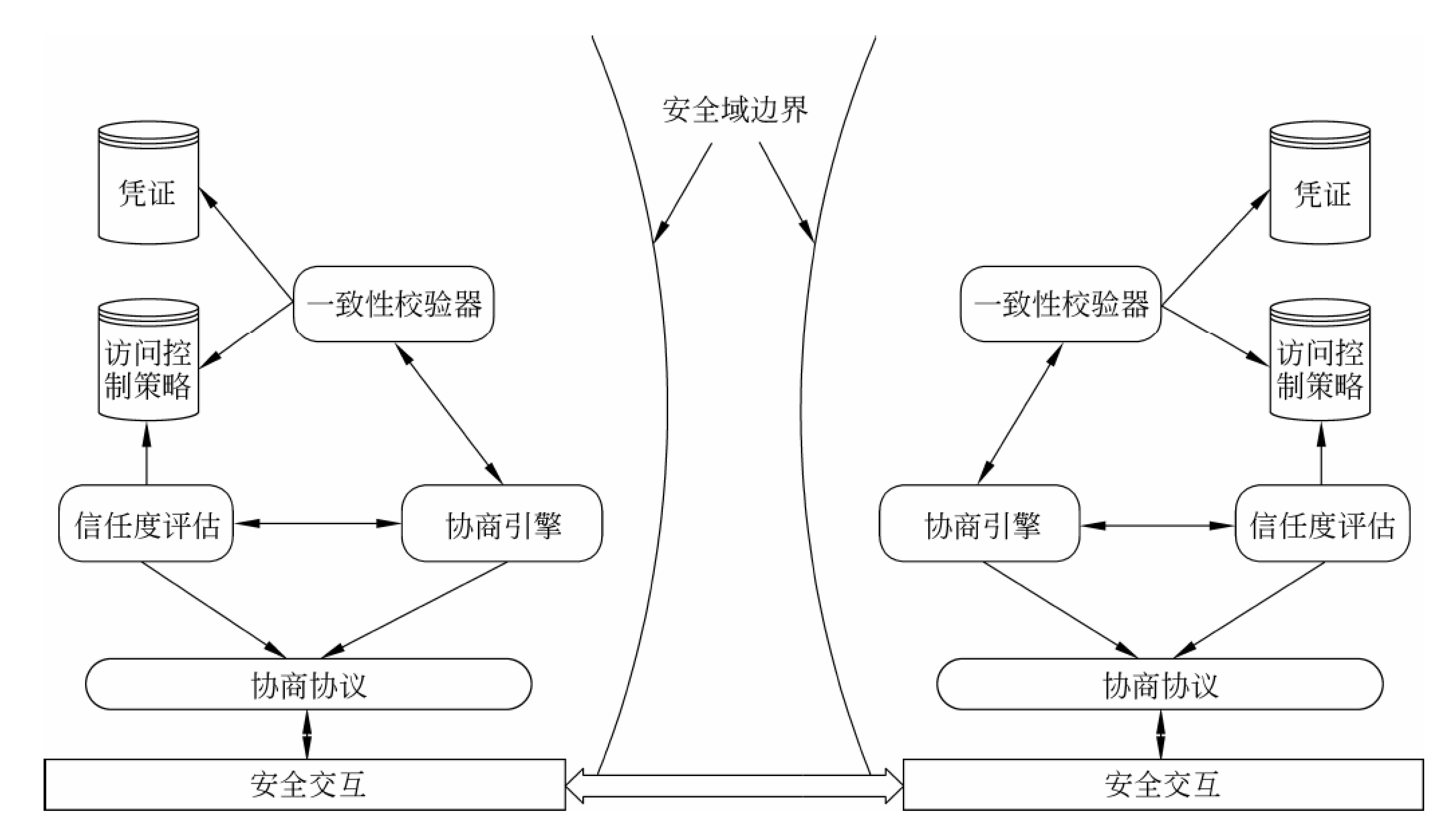


图 4.1　自适应自动信任协商模型框架

信任凭证库中主要包含两种凭证：信任凭证与声明。信任凭证描述信任凭证持有者的身份或一些属性，这些属性由属性对(变量名、值)描述，信任凭证必须经过信任权威机构的认证，具有可验证性和不可伪造性。声明描述信任凭证持有者的一些特别信息，声明不必经过信任权威机构的鉴定，但是声明的使用更加灵活，能更好地规范所提供的服务。本模型中涉及一种特殊的凭证即信任度凭证，即由信任度评估中心颁发的具有评估中心签名的凭证，凭证中包含了某实体的信任度等级信息。

访问控制策略库中包含了凭证的访问控制策略以及服务/资源访问控制策略。当协商策略接收到访问资源请求消息时，需要访问控制策略管理器在访问控制策略库中查找对应的访问控制策略，返回给协商策略模块。协商策略从一致性校验器得到对方要求的信任凭证集合时，同样需要在访问控制策略库中检索这些证书的访问控制策略。

一致性校验器对给定的凭证集合、信任度信息、访问控制策略和请求进行判定，验证凭证集合是否满足访问控制策略，是否能够解锁这个请求。一致性校验器还需要验证凭证的有效性，如果凭证集合中包含证书链，一致性校验器需要构造凭证链。协商策略收到访问控制策略时，将证书策略以及所有的凭证交给一致性校验器，一致性校验器可以找出凭证集合中满足策略的所有子集，并且能够根据凭证敏感性找出代价最小的披露凭证集合。

信任协商会话发起后，协商策略接收会话消息，协调整个信任协商的进行。一个协商策略就是一个函数，它的输入是当前的协商状态，输出是一方应该向另一方显示的下一个信任凭证和访问控制策略的集合。协商策略模块主要负责接收并处理协商主体发送过来的消息，并对其做出响应。协商策略模块接收到请求消息，则处理请求消息，获取对方信任度信息，返回访问控制策略；协商策略接收到访问控制策略消息，则将访问控制策略交给一致性校验模块检索满足策略的凭证集合并返回。协商策略接收到包含凭证的消息，则将凭证集合以及发送给对方的访问控制策略交给一致性校验器，一致性校验器将验证结果返回，协商

策略根据返回的结果对协商做出反馈。如果协商超时或者一方不能提供满足对方访问控制策略的凭证,则终止信任协商。

信任度评估模块主要是对信任协商主体的可信度进行评价,其信任度评估算法在前面章节中已作介绍。系统根据信任度评估算法对信任协商主体的信任度进行初始化和动态调整。服务提供者在收到请求者发出信任协商请求的同时,也收到了请求者的信任凭证。该凭证由信任度评估模块颁发,记录了请求者的信任度信息,以及获取凭证的时间等信息。信任协商策略根据用户的信任度对用户应用适当的访问控制策略。

4.2 自适应自动信任协商工作流程

在信任协商过程中,一般情况下协商双方是为了一个共同的目标而进行信任协商。假设实体 A 为了访问实体 B 的某个资源发起信任协商,那么称 A 为请求方,B 为被请求方。请求方信任协商代理发送协商会话请求,判断被请求方信任协商机制是否同意会话请求,否,则协商失败,不可以访问被请求方的服务或者资源;是,则接受对方消息,并判断对方消息中包含的信息是访问控制策略和信任凭证还是访问资源授权信息,如果是访问资源授权信息,则协商成功,可以访问所述被请求方信任协商机制的服务或者资源;否,则判断对方消息中包含的信息是访问控制策略还是信任凭证,判断是策略,则查找凭证库中是否包含满足访问控制策略的凭证,判断凭证是否受保护,否,则将满足条件的凭证发送给所述被请求方信任协商机制;是,则查看对方信任度,根据信任度查找凭证对应的访问控制策略,将访问控制策略发送给所述被请求方信任协商机制;判断是凭证,则一致性校验器验证凭证,判断是否满足访问控制策略,满足则将解锁的凭证或者访问控制策略,发送给所述被请求方信任协商机制;不满足则协商失败,不可以访问所述被请求方信任协商机制的服务或者资源。

被请求方信任协商机制接收到协商会话请求,同意信任协商,检查会话消息中的请求资源或服务是否存在,不存在则信任协商失败,所述请求方信任协商机制不可访问所述被请求方信任协商机制的服务或者资源;存在则查看请求方的信任度,根据用户的信任度等级在访问控制策略库中查找对应的访问控制策略,并发送访问控制策略。接收所述请求方信任协商机制的消息,查看请求方消息中包含的信息是访问控制策略还是信任凭证,若是策略,则判断策略检索凭证库中是否包含满足访问控制策略的凭证,不包含则信任协商失败,所述请求方信任协商机制不可访问所述被请求方信任协商机制的服务或者资源,包含则判断凭证是否受保护,否,则将满足条件的凭证发送给所述请求方信任协商机制,是则查看对方信任度,将证书对应的凭证访问控制策略发送给所述请求方信任协商机制;若是凭证,则一致性校验器验证凭证是否满足访问控制策略,不满足则信任协商失败,所述请求方信任协商机制不可访问所述被请求方信任协商机制的服务或者资源,满足则判断收到的所有证书是否包含一个可以满足最终资源或者服务访问控制策略的证书集合,否,则将解锁的凭证或者访问控制策略发送给所述请求方信任协商机制,是则协商成功,授予所述请求方信任协商机制访问服务或者资源的权限。

以上所述的自适应自动信任协商是一个相互之间多次披露消息的过程。信任协商中的访问控制策略和凭证披露过程在一次复杂的信任协商中会执行多次。

本节用一个一般的例子来介绍信任协商的工作流程。实例中A和B进行信任协商，A要求访问B的资源R，B获取A的信任度，针对A的信任度，B检索到对R定义的访问控制策略P_r，将P_r发送给A。A在收到B的访问控制策略P_r后，检索证书库，检索到满足策略P_r的证书集合C_{a1}和C_{a2}。证书C_{a2}包含敏感信息，受访问控制策略保护。A获取B的信任度，根据B的信任度检索访问控制策略库，检索到访问控制策略$P_{C_{a2}}$。A将C_{a1}和$P_{C_{a2}}$发送给B。B收到访问控制策略之后检索凭证库，检索到证书C_{b1}满足访问控制策略$P_{C_{a2}}$，并且C_{b1}不受访问控制策略保护，B将C_{b1}发送给A。A验证证书C_{b1}满足访问控制策略$P_{C_{a2}}$，将C_{a2}发送给B。B收到凭证后验证C_{a1}和C_{a2}满足访问控制策略P_r，B授予A访问资源R的权利。整个协商过程可概括如下：

(1) A：向B提出申请，请求访问资源R。

(2) B：将访问控制策略P_r发送给A。

(3) A：请看我的证书C_{a1}和访问控制策略$P_{C_{a2}}$。

(4) B：请看我的证书C_{b1}。

(5) A：请看我的有关C_{a2}凭证。

(6) B：授权访问资源R。

A与B之间信任协商机制的内部流程见图4.2和图4.3。

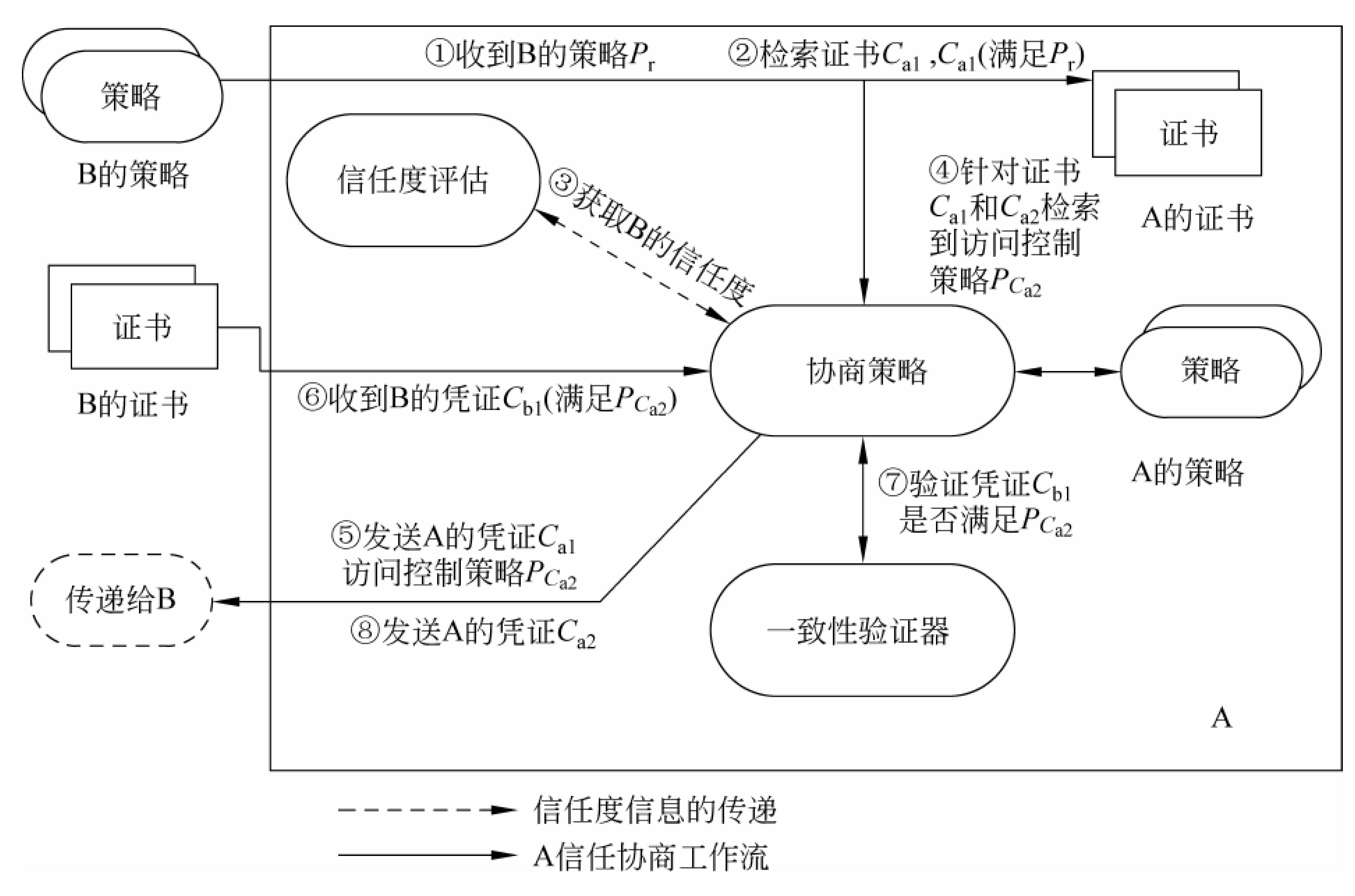


图4.2 实体A信任协商的内部工作流程

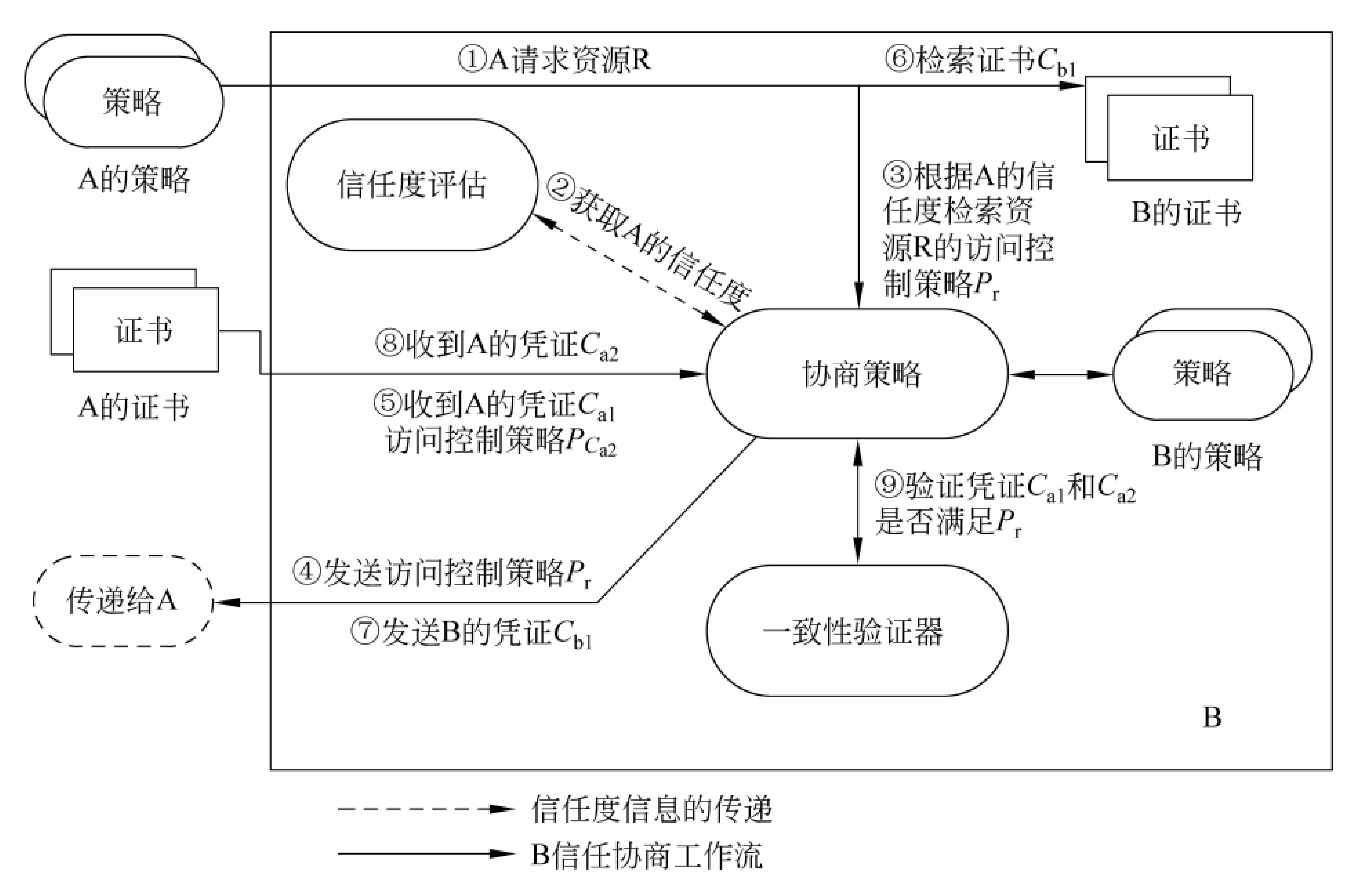


图 4.3 实体 B 信任协商的内部工作流程

4.3 自适应策略模式及分析

4.3.1 自适应策略模式

本节通过信任协商中一个典型案例来说明自适应自动信任协商策略。为了方便说明问题,假设一次消息只能传递一个访问控制策略或者信任凭证。

两位在线书店代理商 Bob 与 Candy 是商业合作伙伴关系,也就是说 Bob 书店的会员与 Candy 书店的会员可在这两个书店享受打折优惠。顾客 Alice 是 Candy 的会员,她想在 Bob 的书店以打折价买书。Bob 要求 Alice 出示有效会员卡及信用卡号。由于信用卡带有隐私信息,Alice 只能向经过 BBB(信誉授权机构)认证的组织公开信用卡号[1]。采用吝啬策略的协商过程如图 4.4 所示。

采用热心策略,协商过程如图 4.5 所示。

下面介绍采用自适应信任协商策略的协商过程。

若 Bob 书店与多个用户成功交易,信誉较好,假设 Bob 的信任度等级(TG)为 3。Alice 的信任度等级为 0。Alice 根据 Bob 的信任度调整访问控制策略,不需要验证 Bob 的“Candy 合作伙伴属性”凭证。协商过程如图 4.6 所示。

随着 Bob 书店的生意越来越好,完善的服务使 Bob 书店的信誉度提升,在信任协商过程中 Bob 的信任度也提高,信任等级达到非常高的级别。假设这时 Bob 的信任等级为 5,Alice 的信任等级为 0。那么 Alice 相应地对访问控制策略进行调整,Bob 不用出示任何凭证,Alice 将所需的凭证直接发送给 Bob。协商过程如图 4.7 所示。

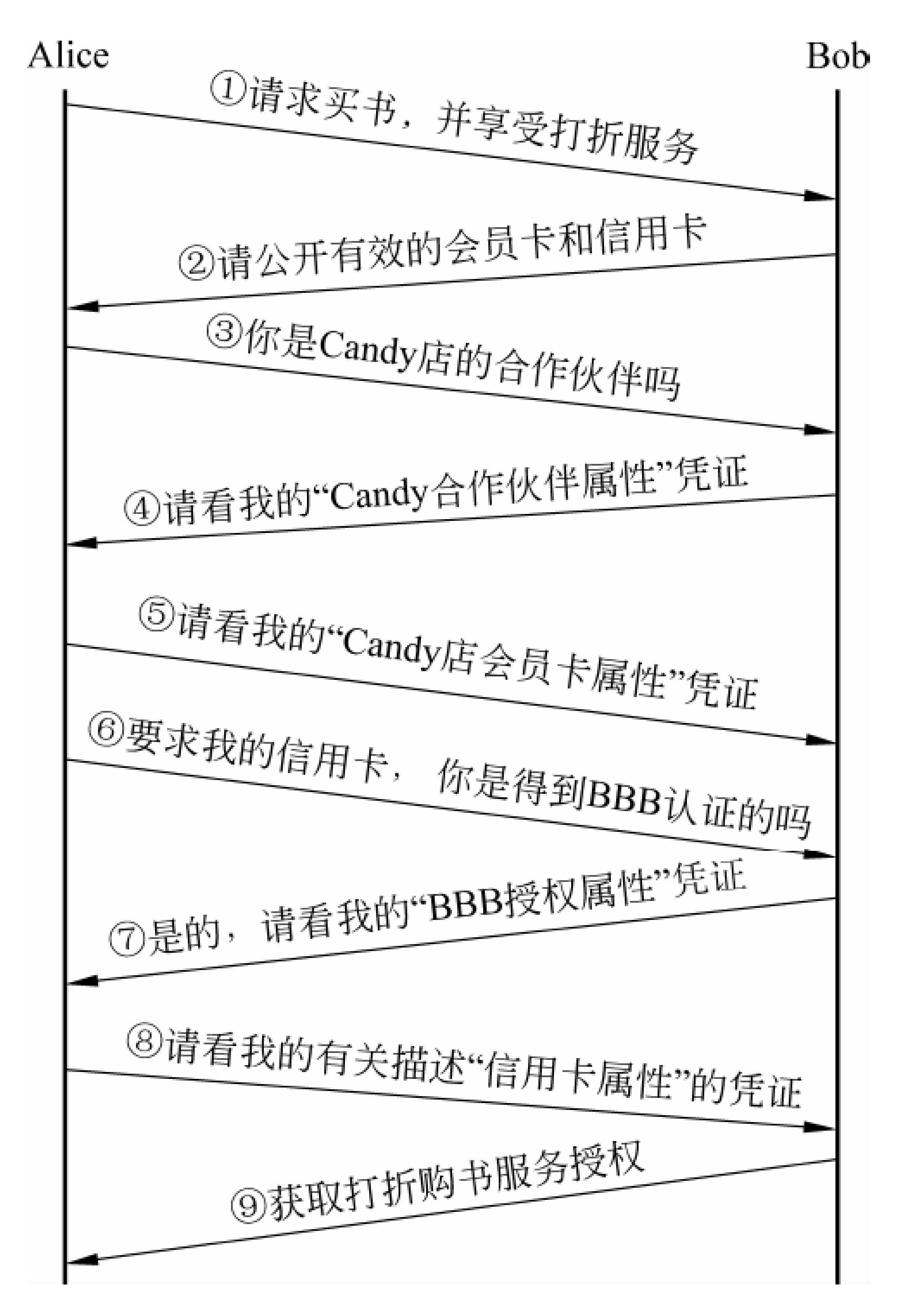


图 4.4　吝啬策略的信任协商过程

图 4.5　热心策略的信任协商过程

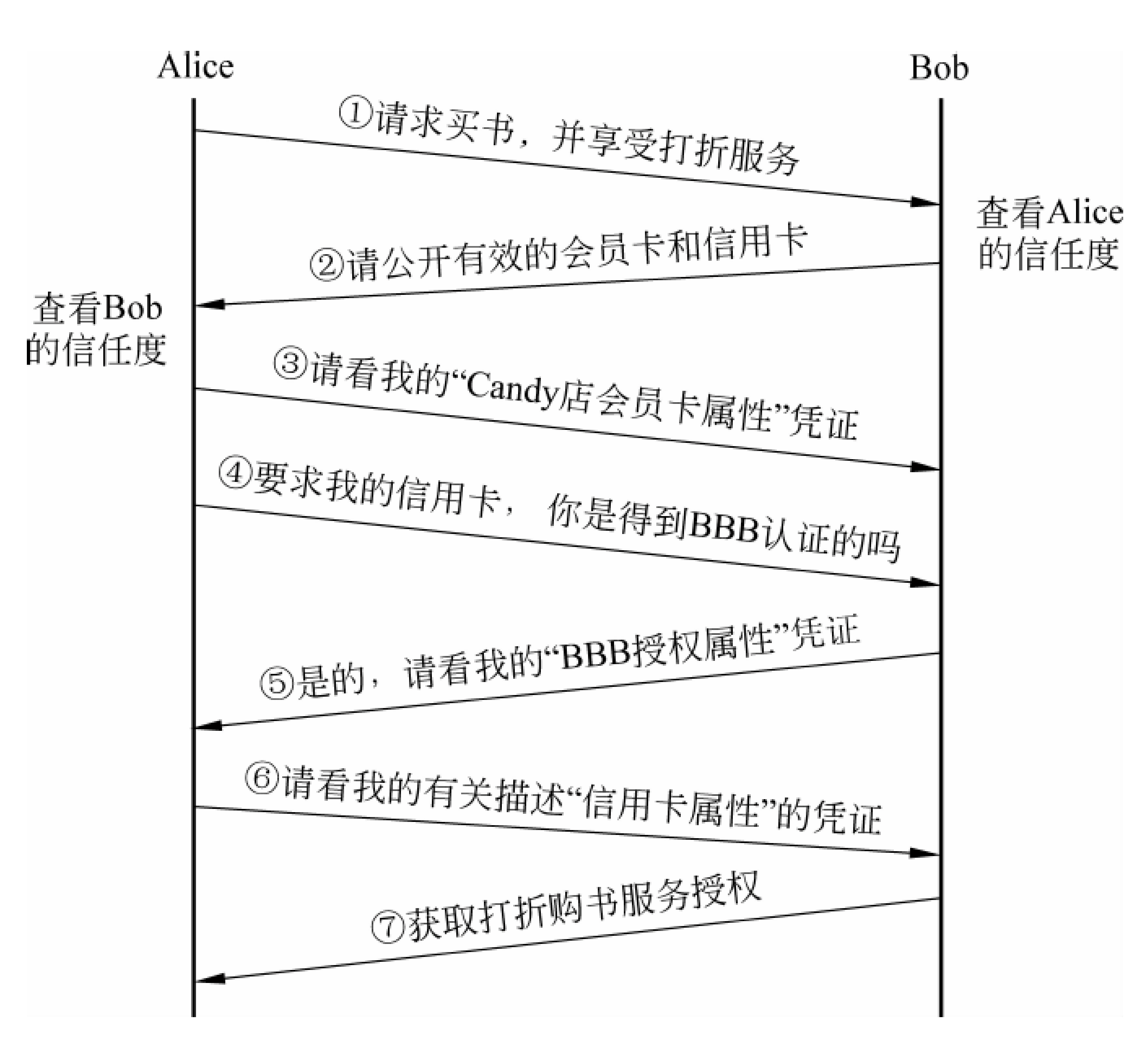


图 4.6　自适应信任协商流程 1

Alice 通过经常光顾 Bob 的书店积累了较高的信誉，这时 Bob 书店对 Alice 也更加信任了。假设这时 Bob 的信任等级为 5，Alice 的信任等级为 3。双方进行协商时，基于双方信任度较高，不需要再出示凭证就可直接使 Alice 享受打折服务。协商过程如图 4.8 所示。

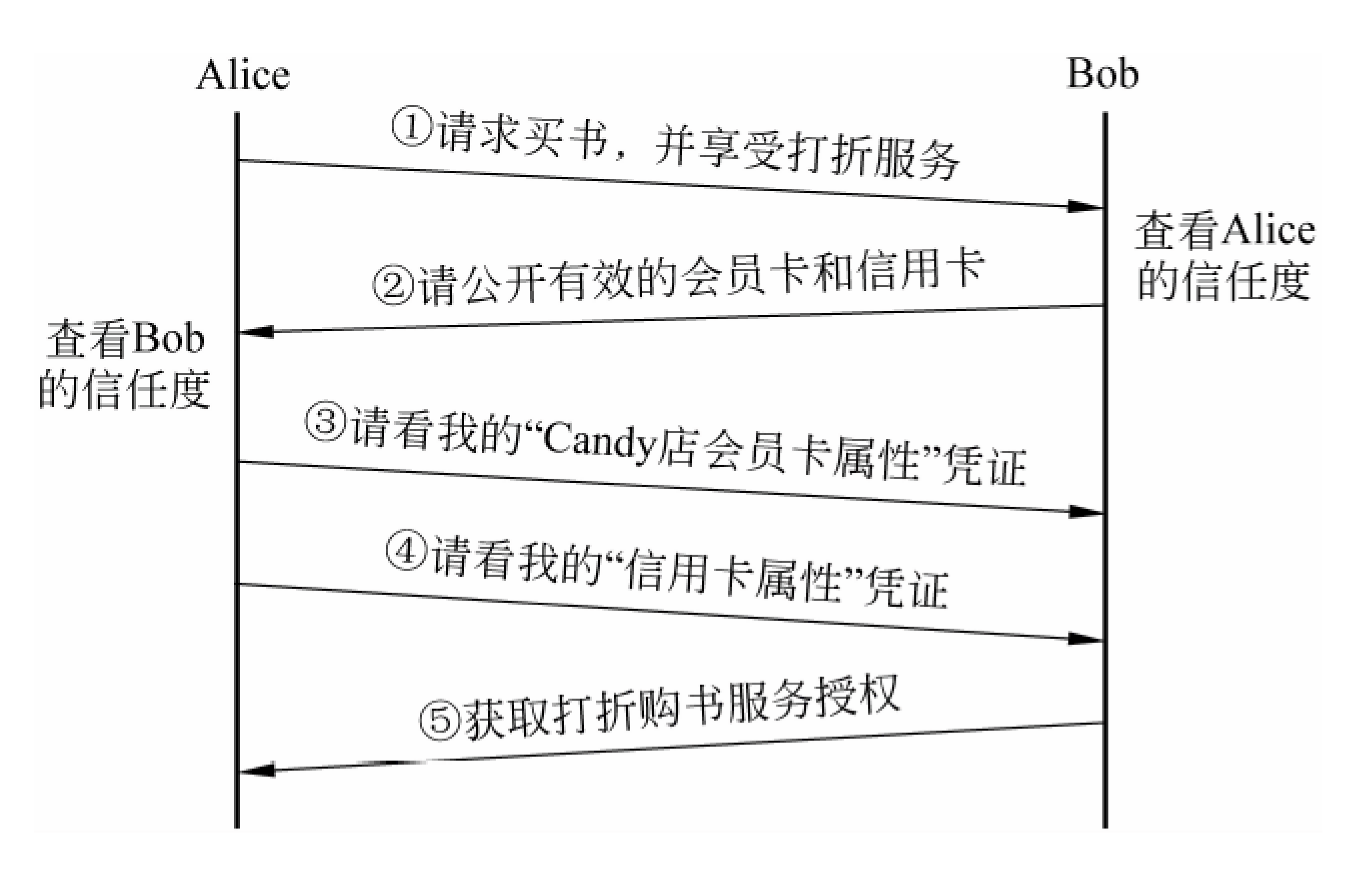


图 4.7　自适应信任协商流程 2

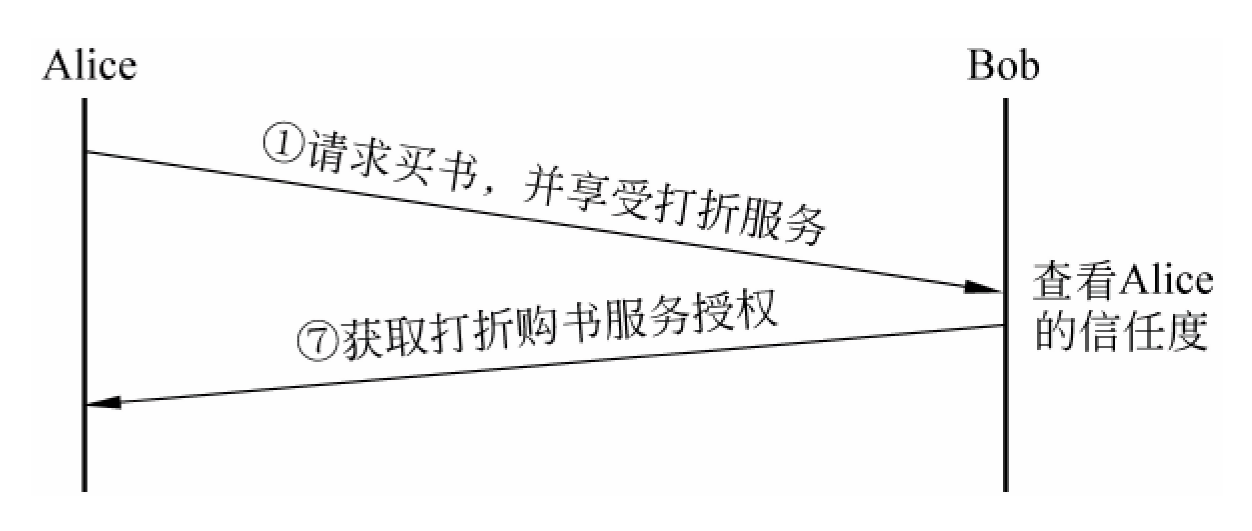


图 4.8　自适应信任协商流程 3

4.3.2　实验分析

一次成功信任协商的时间主要由证书和策略交换时间、证书发现与检索时间以及一致性校验时间组成。表 4.1 列出了上述信任协商过程中的时间开销对比数据。

表 4.1　信任协商过程中时间开销的对比

协商策略/事件发生次数	信任度查看	凭证策略检索	凭证策略披露	一致性校验
吝啬策略	0	4	4	3
热心策略	0	1	3	2
自适应协商策略	—	—	—	—
Bob：TG＝－1；Alice：TG＝－1	2	4	4	3
Bob：TG＝0；Alice：TG＝0	2	4	4	3
Bob：TG＝3；Alice：TG＝0	2	4	4	2
Bob：TG＝5；Alice：TG＝0	2	2	2	1
Bob：TG＝5；Alice：TG＝3	1	0	0	0

另外，我们采用两台相同配置的 PC 作为 Bob 和 Alice，来模拟打折服务。比较热心策略、自适应策略和吝啬策略在协商效率上的差别，针对上述协商过程编写相应的信任协商策略，记录各个协商过程的时间。实验结果如图 4.9 所示。从图 4.9 中可以看出，在用户信任度较低，接近初始信任度时，吝啬策略与自适应协商策略开销时间差别不大，热心策略开销较小；随着用户信任度的提高，自适应协商策略开销时间逐渐缩短，趋向于热心策略开销时间；当用户信任度达到一定程度时，不需要交换凭证和策略，直接访问资源，自适应协商策略

的开销时间最短。信任协商过程越复杂，自适应协商策略的优势就越明显。用户信任度越高，信任协商过程越简单，信任协商时间越短。

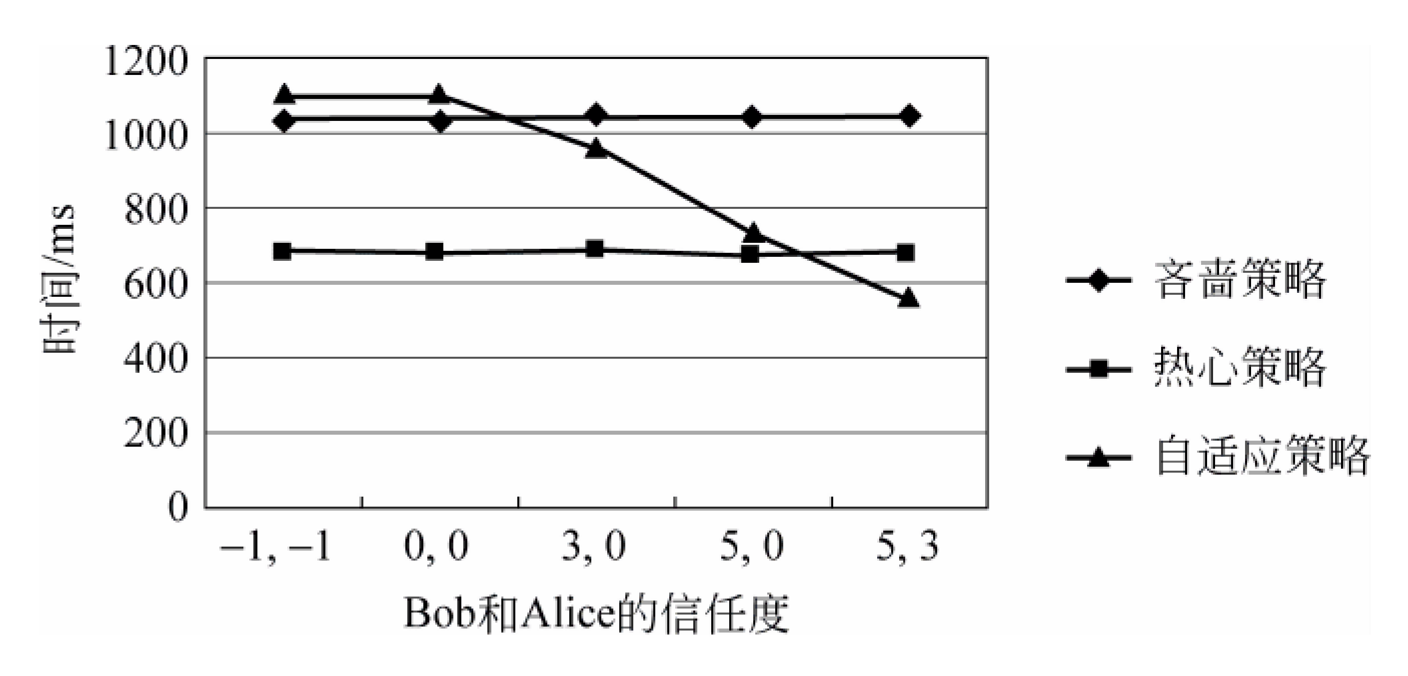


图 4.9　热心策略、自适应策略和吝啬策略的对比

4.4　一致性校验器

一致性校验器是 AATN 模型中的一个重要组成部分，其中一致性校验算法更是所有自动信任协商系统的核心要素。一致性校验器可以判断给定的信任凭证集合是否能够满足针对请求资源定义的本地策略。如果将一致性校验器看作一个函数，它的输入是信任凭证集合、访问控制策略以及对资源的访问请求，它的输出是判定结果和引导信息。它将判定的结果反馈到协商策略模块。一致性校验器在验证证书是否满足访问控制策略时，首先验证信任凭证的有效性，在进行匹配时过滤掉无效的证书。一致性校验器的另一个重要功能就是针对给定的访问控制策略检索本地满足策略的信任凭证集合。信任协商是信任凭证和访问控制策略的披露过程。协商策略在对策略模式进行选择后，由一致性校验器来进行后续的工作。

自动信任协商对一致性校验器提出了更高的要求，自动信任协商要求协商双方在一致性校验失败时，即对方提供的信任凭证不能满足本地的访问控制策略时，给予对方有价值的反馈信息以引导信任协商的进行，由于满足一条访问控制策略的信任凭证集合往往不止一个，所以当一方提供的凭证集合不能满足对方的访问控制策略时，并不代表此次协商没有成功的路径，因此这时需要给提供凭证的一方有价值的反馈信息才能引导信任协商的继续进行。

目前已经存在一些针对特定系统的一致性校验算法。信任管理系统如 PolicyMaker 和 KeyNote 等提出了适应自身系统的一致性校验算法。在这些信任管理系统中，一致性校验器提供了验证一个信任凭证集合是否可以满足特定访问控制策略的功能。

PolicyMaker 中实现的一致性校验算法需要解决诸如断言的调用顺序、断言的调用次数和丢弃产生冲突的断言等一系列问题[2]。KeyNote 的一致性校验算法是一种深度优先算法，其主要思想是采用递归的方式试图查找到至少一条能够满足请求的策略断言[3]。REFREE 能够在一致性证明验证时自动收集并验证安全信任凭证的可靠性，应用系统仅需给出初始的安全策略、安全信任凭证和验证内容以及一些必要的验证上下文信息[4]，这一点有利于该信任管理系统的使用。自动信任协商系统如 TrustBuilder 也提出了适合自身的一些一致性校验器算法，但在信任协商失败信息反馈上仍存在不足。

在探讨一致性校验器之前，需先对访问控制策略的内部逻辑结构进行分析。

4.4.1 访问控制策略描述

1. 访问控制策略的内部结构

访问控制策略由若干条规则组成，也可以称为子策略。本章中用 C 表示信任凭证，Policy 表示访问控制策略。

定义 4.1 规则(rule)：规则描述了对一个凭证集合的约束，其中包括对每个凭证属性信息的约束。一条规则可以作用于一个或者多个凭证对象。

为了表述方便作出如下定义：

Policy(R,TG,rules)：代表一条访问控制策略，Policy 代表策略名称；R 代表访问控制策略保护的资源，R 可以是资源或服务、信任凭证以及访问控制策略；TG 为本策略所适用的用户信任等级，是一个信任等级范围；rules 代表若干条规则，这些子规则之间可以是“与”、“或”的逻辑关系。

rules($rule_0$, $rule_1$, …, $rule_n$)：$rule_0$, $rule_1$, …, $rule_n$ 之间的逻辑关系用 ∧ 或 ∨ 符号连接，表示“与”、“或”的逻辑关系。为了简化算法，将逻辑“与”关系的子规则放到 rules 的前面，包含复杂关系的规则放到 rules 的后面。例如

$$rule_0 \wedge rule_1 \wedge (rule_2 \vee rule_3)$$

rule(Constraint(C_0, C_1, …, C_n))：表示一条规则，此条规则包含对信任凭证 C_0、C_1、…、C_n 的描述。信任凭证 C_0、C_1、…、C_n 存在两种组合关系，一种是此凭证集合组成一条凭证链，一种是此凭证集合之间存在“与”、“或”逻辑关系。

定义 4.2 链规则(chain-rule)：一条规则中描述了对一个凭证集合的约束，如果这个凭证集合中的凭证形成了若干个凭证链，那么称这条规则为链规则。凭证链的建立是通过 A 给 B 颁发证书，而 B 又以此证书为签名给 C 颁发证书，这样就形成了凭证链。

例如，$Policy_0$(R_0, TG, $rule_a$)是一条访问控制策略的表达式。访问控制策略的名称是 $Policy_0$，策略是对资源 R_0 定义的访问控制策略。

$rule_a$(Constraint(C_1-C_2-C_3))：$rule_a$ 表示规则的名称，此条规则定义了对信任凭证 C_1、C_2 和 C_3 的约束，并且 C_1、C_2 和 C_3 形成了一条信任凭证链。称这条访问控制策略为链策略。

定义 4.3 混合规则(mixing-rule)：如果一条规则中描述的凭证集合既包含独立的凭证又包含凭证链，那么称这条规则为混合规则。

例如，$Policy_1$(R_1, TG, $rule_a \vee rule_b$)是一条访问控制策略的表达式。访问控制策略的名称是 $Policy_1$，策略是对资源 R_1 定义的访问控制策略。

$rule_a$(Constraint(C_1-C_2-C_3))：$rule_a$ 表示子规则的名称，此条规则定义了对信任凭证 C_1、C_2 和 C_3 的约束，并且 C_1、C_2 和 C_3 形成了一条信任凭证链。

$rule_b$(Constraint(C_4, C_5, C_6))：$rule_b$ 表示规则的名称，此条规则定义了对信任凭证 C_4、C_5 和 C_6 的约束，并且 C_4、C_5 和 C_6 之间是逻辑“与”、“或”的关系，但不构成信任凭证链。

称这条访问控制策略为混合策略。

2. 访问控制策略树

通过对访问控制策略内部结构的研究，我们将访问控制策略构造成信任凭证有向树。

通过对访问控制策略树的构造、遍历与标记过程，可以找出满足访问控制策略的所有信任凭证集合以及最小代价信任凭证集合。在验证信任凭证集合是否满足访问控制策略的过程中，本节给出了一致性校验失败信息反馈的方法。

根据上一节对访问控制策略的描述，我们通过对例 4.1 的分析来介绍访问控制策略树的构造过程。为了简化访问控制策略树，将只关系到单个凭证和凭证链的合取关系放至访问控制策略的前端，多个凭证的合取析取则放至访问控制策略的后端。例 4.1 的访问控制策略树如图 4.10 所示。

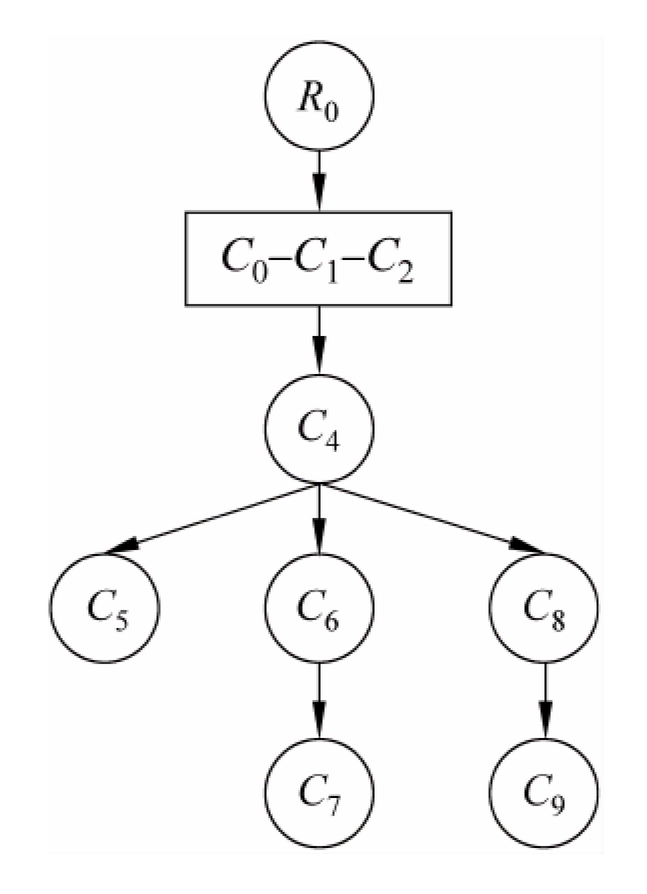


图 4.10　访问控制策略树

例 4.1　$Policy_0(R_0, TG, rule_a \wedge rule_b \wedge rule_c)$是一条访问控制策略的表达式。访问控制策略的名称是 $Policy_0$，是对资源 R_0 定义的访问控制策略。TG 的访问控制策略范围是：$5>TG>2$。

$rule_a(Constraint(C_1\text{-}C_2\text{-}C_3))$：$rule_a$ 表示规则的名称，此条规则定义了对信任凭证 C_1、C_2 和 C_3 的约束，并且 C_1、C_2 和 C_3 形成了一条信任凭证链。

$rule_b(Constraint(C_4))$：$rule_b$ 表示规则的名称，此条规则定义了对信任凭证 C_4 的约束，$Constraint(C_4)=C_4$。

$rule_c(Constraint(C_5, C_6, C_7, C_8, C_9)$：$rule_c$ 表示规则的名称，此条规则定义了对信任凭证 C_5、C_6、C_7、C_8 和 C_9 的约束，并且 C_5、C_6、C_7、C_8 和 C_9 之间是逻辑“与”、“或”的关系，但不构成信任凭证链。$Constraint(C_5, C_6, C_7, C_8, C_9)=C_5 \vee (C_6 \wedge C_7) \vee (C_8 \wedge C_9)$。

由以上描述可知此访问控制策略是一条混合策略。

这里规定，在以上的规则之间以及凭证之间不允许定义如$((C_0 \vee C_1) \wedge (C_2 \vee C_3))$这样的关系，而应将此种关系转化为$(C_0 \wedge C_2) \vee (C_0 \wedge C_3) \vee (C_1 \wedge C_2) \vee (C_1 \wedge C_3)$表示。即在构造访问控制策略树之前，将前者转化为后者的表示方式。

定义访问控制策略树的根节点为资源节点，表示访问控制策略保护的资源。将与资源直接相关的单个凭证和凭证链的合取节点直接作为父节点的唯一孩子节点依次连接到树的节点上。对于析取关系的节点并列作为父节点的孩子节点连接到访问控制策略树上。对于每条规则进行以上操作，按照约定将合取节点写在析取节点的前面，所以每次操作都会先将路径中必需的凭证作为父节点的孩子节点连接到访问控制策略树上。在访问控制策略树中用顺序孩子节点表示合取关系，用父节点的多个孩子节点表示析取关系。

4.4.2　一致性校验算法

根据访问控制策略树的构造过程可知，访问控制策略中根节点的孩子节点，以及孩子节点中的唯一孩子节点是访问控制策略中定义的必要信任凭证或凭证链。在合取节点内部用同样的方法构造访问控制策略。由以上分析可知，访问控制策略树中每条从根节点到叶子节点的路径都是一个能够解锁资源的信任凭证集合。

1. 访问控制策略树的标记

对访问控制策略树的标记，按照深度优先的算法从根节点进行遍历。每个节点有这样的存储结构：(父节点，到根节点的距离，凭证敏感度，综合敏感度)。假设凭证受访问控制策略保护，即敏感度为 1，否则敏感度为 0。

(1) 如果根结点只有一个孩子节点,那么检索本地凭证集合中是否存在此孩子节点,并标记孩子节点的父节点,到根节点的距离等于1,标记此凭证敏感度和综合敏感度。如果本地信任凭证集合中不存在此节点的凭证,那么访问控制策略树标记结束,不存在满足访问控制策略的信任凭证集合。

(2) 如果根节点有多个孩子节点,那么按照深度优先的方式遍历访问控制策略树,标记孩子节点的父节点、到根节点的距离等于1、凭证敏感度和综合敏感度。如果本地的信任凭证中不存在此节点的凭证,那么终止此路径的遍历,依次回溯,直到找到某节点存在右相邻的兄弟节点,标记以此节点的右侧兄弟节点为根节点的子树。如果不存在右相邻的兄弟节点,则标记算法结束。

(3) 如果根节点的孩子节点为信任凭证链,那么检索信任凭证库中是否存在所有信任凭证,标记该信任凭证链的父节点、到根节点的距离和综合敏感度。如果信任凭证不存在,那么回溯,直到找到一个节点中存在右相邻的兄弟节点,对以此右相邻的兄弟节点为根结点的子树进行标记。如果不存在这样的节点,则标记结束。

(4) 对根节点的孩子节点进行相同的操作,如果此节点只有一个孩子节点,那么标记孩子节点的父节点、到根节点的距离(即父节点到根节点的距离加1)和信任凭证敏感度,并标记综合敏感度为父节点的敏感度加上此节点的敏感度。如果本地的信任凭证中不存在此节点的凭证,那么终止此路径的遍历,依次回溯,直到节点的父节点存在多个孩子节点,并且仍有右侧相邻的兄弟节点,对此右侧相邻的兄弟节点进行深度优先标记,否则标记算法结束。

(5) 如果正在遍历的节点为叶子节点,那么标记本节点的父节点、到根节点的距离、凭证敏感度和综合敏感度等,并记录本节点的信息。到此找到了一个信任凭证的暴露集合。

(6) 对访问控制策略树进行以上的标记过程,直到整个遍历过程结束。如果遍历到的叶子节点集合不为空,那么叶子节点的个数即为满足访问控制策略的集合数。比较叶子节点中的综合敏感度,选择综合敏感度最小的叶子节点或叶子节点集合。如果存在多个综合敏感度最小的叶子节点,那么找出综合敏感度最小的叶子节点中到根节点距离最小的叶子节点。

根据例4.1给出访问控制策略树的标记过程如下:

(1) 首先判断根节点的孩子节点,得根节点的唯一孩子节点为信任凭证链 C_1-C_2-C_3,检索是否包含信任凭证。假设本地包含此信任度凭证链,且此信任凭证链不存在敏感度凭证,那么标记节点(C_1-C_2-C_3)的信息:父节点、到根节点的距离、凭证敏感度和综合敏感度为(R_0,1,0,0)。

(2) 标记(C_1-C_2-C_3)的孩子节点 C_4。假设本地包含信任凭证 C_4,且 C_4 为敏感度凭证,C_4 到根节点的距离为2,凭证敏感度为1,综合凭证敏感度为1。标记为 C_4((C_1-C_2-C_3),2,1,1)。

(3) 按照深度优先的顺序标记 C_4 的孩子节点。首先标记第一个孩子节点 C_5,假设本地包含信任凭证 C_5,且 C_5 为敏感度凭证。C_5 到根节点的距离为3,凭证敏感度为1,综合凭证敏感度为2。标记为 C_5(C_4,3,1,2)。

(4) C_5 为叶子节点,此条路径遍历结束,回溯遍历 C_5 的兄弟节点 C_6。假设本地包含信任凭证 C_6,且 C_6 为敏感度凭证。标记为 C_6(C_4,3,1,2)。

(5) 假设存在 C_7、C_8 和 C_9,且其敏感度分别为1、0和1,则标记为 C_7(C_6,4,1,3),C_8(C_4,3,0,1),C_9(C_8,4,1,2)。

根据上面的分析得到图 4.11，从图 4.11 中可以看出存在 3 个信任凭证集合可以满足访问控制策略，分别是（C_1-C_2-C_3，C_4，C_5），（C_1-C_2-C_3，C_4，C_6，C_7），（C_1-C_2-C_3，C_4，C_8，C_9）。根据叶子节点的综合敏感度可知（C_1-C_2-C_3，C_4，C_5）和（C_1-C_2-C_3，C_4，C_8，C_9）为综合敏感度最小的信任凭证集合。根据叶子节点与根节点的距离可知（C_1-C_2-C_3，C_4，C_5）披露的凭证最少，所以选择披露信任凭证集合（C_1-C_2-C_3，C_4，C_5）。

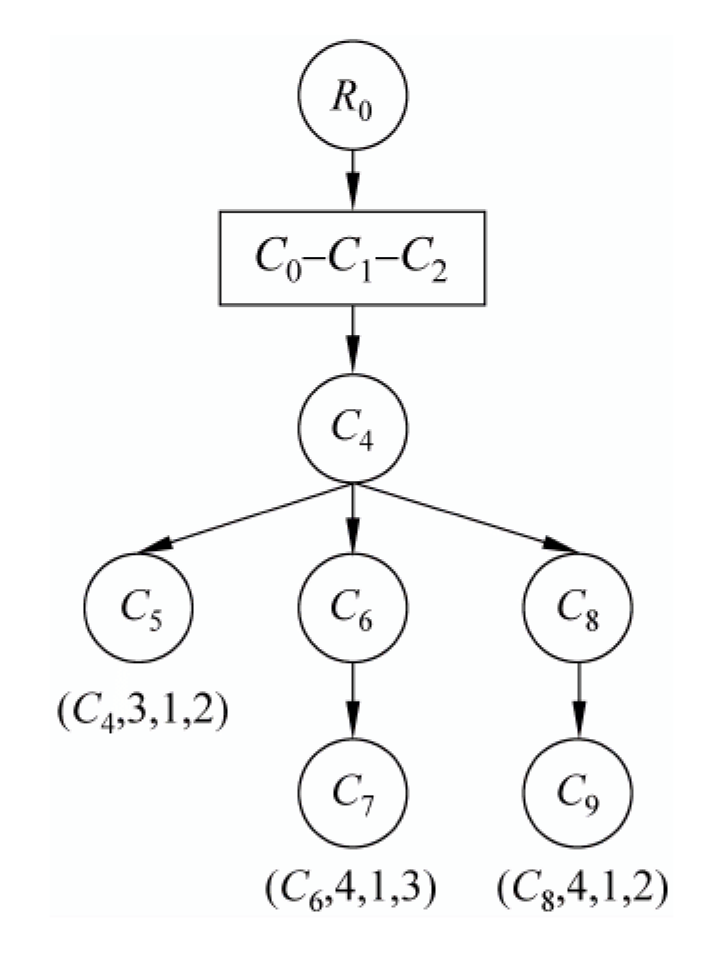


图 4.11　访问控制策略标记树

2. 信任协商失败信息反馈

通过信任凭证的验证过程，可以给出验证失败的信息。一致性校验算法的一个主要功能就是验证一个信任凭证集合是否满足访问控制策略，返回结果并给出有助于进一步协商的信息。

假设在例 4.1 中 A 与 B 进行信任协商，且 B 将以上访问控制策略发送给 A，根据以上的算法找出了本地满足访问控制策略的集合，经过若干次访问控制策略与信任凭证的交互暴露过程，A 将综合敏感度最小、数量最小的访问控制策略集合（C_1-C_2-C_3，C_4，C_5）发送给 B。然后 B 的协商模块将会把 A 的请求消息、A 发送的信任凭证集合以及 B 的访问控制策略交给一致性校验器。一致性校验器对其进行验证。

（1）首先根据访问控制策略树的根节点与 A 的请求消息判断此访问控制策略是否对本请求进行保护。如果是，则对访问控制策略树进行广度优先遍历。

（2）将 A 的信任凭证集合进行验证，验证信任凭证的有效性。验证证书是否过期、是否注销等，如果信任凭证均有效则转到（3），否则抛弃无效凭证，信任协商失败信息置为存在无效凭证，转到（3）。

（3）如果访问控制策略树的根节点有唯一孩子节点，那么用 A 的信任凭证匹配此节点的信任凭证，查看是否存在此节点的信任凭证，是否满足策略定义的属性条件。如果满足则继续验证此节点的孩子节点。否则验证失败，信任协商失败信息置为缺少必要凭证。

（4）如果访问控制策略树的根节点有多个孩子节点，那么依次从左向右验证根节点的孩子节点，如果存在信任凭证满足某个孩子节点的约束，对此节点进行广度优先遍历。如果对此节点的广度优先遍历并未找到满足约束条件的路径，那么返回遍历根节点的其余孩子节点。如果不存在满足根节点的信任凭证满足某孩子节点的约束，那么验证过程结束。信任协商失败信息置为缺少必要凭证。

（5）如果访问控制策略树中根节点的孩子节点是一个链节点，那么对此节点的约束的信任凭证链进行验证。首先检索是否存在此链接点的根凭证，如果存在并且满足约束，那么检索凭证链中的下一个凭证。依次进行检索匹配，如果信任凭证链中的所有凭证均存在，那么构造此凭证链。如果某个信任凭证不存在，那么此节点验证失败。信任协商失败信息置为信任凭证链构造失败。

（6）对根节点的孩子节点进行（3）、（4）、（5）的操作。

（7）如果找到一个节点为叶子节点，且存在满足此叶子节点的信任凭证，那么遍历结束，验证成功。一致性校验器将结果反馈到协商模块，允许访问请求的资源。

4.4.3 完备性分析

一致性校验器是信任协商的核心部件，应该具有完备性。即对于一个以某条访问控制策略、信任凭证集合和请求为输入的一致性校验过程来说：

(1) 如果此信任凭证能够满足访问控制策略，并且信任凭证均有效，那么一致性校验器验证结果一定为真；反之，如果一致性校验结果为真，那么这个信任凭证集合一定能够满足这个访问控制策略。

(2) 如果此信任凭证不能够满足此访问控制策略，那么一致性校验结果一定为假；反之，如果一致性校验结果为假，那么此信任凭证集合一定不能满足此访问控制策略。

访问控制策略树的构造过程就是寻找信任凭证集合的过程，每个从根节点到叶子节点的所有信任凭证集合都是一个满足访问控制策略的信任凭证集合。通过对访问控制策略树的遍历匹配过程，如果存在一个信任凭证集合满足这个访问控制策略，那么一定能够匹配到一个叶子节点找到这样的路径。

证明：对于访问控制策略 Policy(R，TG，$C_1 \wedge C_2 \wedge C_3$)，根据访问控制策略树的构造方法，将“与”关系的信任凭证作为当前节点的唯一孩子节点添加到访问控制策略树中。将 C_1、C_2、C_3 作为 R 节点的唯一孩子结点依次添加到访问控制策略树中，即 C_1 作为 R 的唯一孩子节点，C_2 作为 C_1 的唯一孩子节点，C_3 作为 C_2 的唯一孩子节点添加到访问控制策略树中。

访问控制策略树构造如图 4.12 所示。

通过对根节点的深度优先遍历，必经过结点 C_1、C_2 和 C_3，如果本地存在信任凭证 C_1、C_2 和 C_3，一致性校验算法一定可以遍历到叶子节点 C_3，找到路径(R-C_1-C_2-C_3)能够满足此访问控制策略。

对于访问控制策略 Policy(R，TG，$C_1 \vee C_2 \vee C_3$)，根据访问控制策略树的构造方法，将“或”关系的信任凭证作为当前节点的孩子节点并列添加到访问控制策略树中，即 C_1、C_2 和 C_3 作为 R 节点并列的子节点。

访问控制策略树构造如图 4.13 所示。

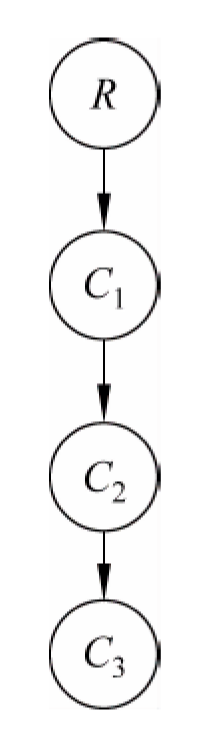


图 4.12 “与”关系访问控制策略树

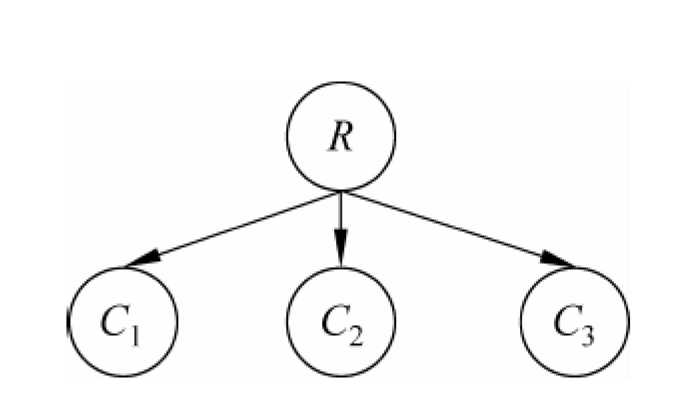


图 4.13 “或”关系访问控制策略树

通过根节点的深度优先遍历算法，如果本地存在 C_1、C_2 和 C_3 中的任何一个信任凭证，一致性校验算法一定可以遍历到某一叶子节点。找到一条或多条路径(R-C_x)满足此访问控制策略。

对于访问控制策略 Policy(R,TG,$C_1 \vee (C_2 \wedge C_3)$)根据访问控制策略的构造方法,将“或”关系节点作为当前节点的并列孩子节点添加到访问控制策略树中,将“与”关系节点作为当前节点的唯一孩子节点添加到访问控制策略树中。C_1 作为 R 的第一个孩子节点,C_2 作为 R 的第二个孩子节点,C_3 作为 C_2 的唯一孩子节点。

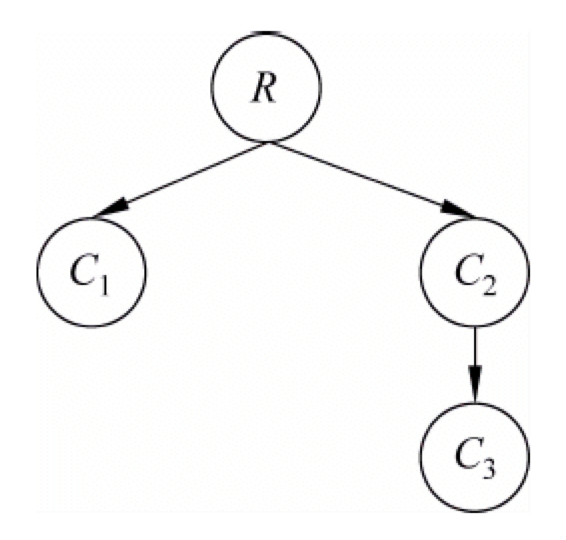


图 4.14 混合关系访问控制策略树

访问控制策略树构造如图 4.14 所示。

通过根节点的深度优先遍历算法,如果本地存在 C_1 或 C_2、C_3 中的任何一组信任凭证集合,一致性校验算法一定可以遍历到某一叶子节点。找到一条或多条路径(R-C_1)或(R-C_2-C_3)满足此访问控制策略。

4.5 本章小结

自动信任协商为所有基于开放、分布、动态特性环境的安全和信任问题提供了新的解决方法。本章提出了一种能根据协商方信任度的计算结果动态调整访问控制策略和协商策略的自适应自动信任协商(AATN)模型。该模型能提供一种灵活高效的安全信任协商机制,对于信任度高的协商方可提供快速响应的策略,而对于信任度低的协商方则提供更为谨慎安全的策略。本章阐述了该模型框架和工作流程,重点讨论了自适应策略模式和一致性检验器等关键技术。另外,通过实验分析,论证了 AATN 所采用的自适应策略模式在兼顾效率和安全两方面需求的有效性。

参考文献

[1] M. Winslett, Yu T, K. E. Seamons, et al. Negotiating trust on the web [J]. IEEE Internet Computing,2002,6(6):30-37.

[2] M. Blaze,J. Feigenbaum,M. Strauss. Compliance Checking in the Policy Maker Trust Management System. In Financial Cryptography,British West Indies,Feb. 1998.

[3] Blaze, M., Feigenbaum, J., Keromytis, A. D. Keynote: trust management for public-key infrastructures. In:Christianson,B.,Crispo,B.,William,S.,et al.,eds. Cambridge 1998.

[4] Chu Yang Hua,Joan Feigenbaum,Brian LaMacchia,et al. REFEREE: Trust Management for Web Applications[J]. World Wide Web Journal,1997,2(2):127-139.

[5] Winsborough WH,Seamons KE,Jones VE. Automated trust negotiation. In: DARPA Information Survivability Conf. and Exposition. New York:IEEE Press,2000:88-102.

[6] Wenbao Jiang, Shaoxu Guo, Wenliang Chen. A Trust Evaluation Model and Algorithm Based on Network Behavior Detection. In: 2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology,October 2010.

[7] Shaoxu Guo,Wenbao Jiang. An Adaptive Automated Trust Negotiation Model and Algorithm [J]. In: International Conference on Communications and Intelligence Information Security. Nanning, Guangxi Province,China: IEEE Press,2010:130-134.

[8] 郭少旭.自适应信任协商技术研究.北京信息科技大学硕士学位论文,2010.

第 5 章　自适应信任协商系统设计

基于第 4 章提出的自适应自动信任协商模型，本章主要讨论如何设计和实现一个自适应信任协商系统。

5.1　系统总体设计

自适应信任协商系统的主要内容包括证书管理器、策略管理器、协商决策模块、信任度评估模块和一致性校验器等，是对自动信任协商系统的改进。证书管理器主要负责管理加载证书和声明；策略管理器主要负责管理访问控制策略；信任度评估模块主要对用户的可信度做出评价，并提供可验证的功能；一致性校验器接收决策模块的请求，判断凭证集合是否满足访问控制策略；证书链处理模块验证凭证集合是否构成信任凭证链；协商决策模块调用其他各个模块以获得协商过程中需要的中间结果，负责整个信任协商过程的协调。该系统的模块图如图 5.1 所示。

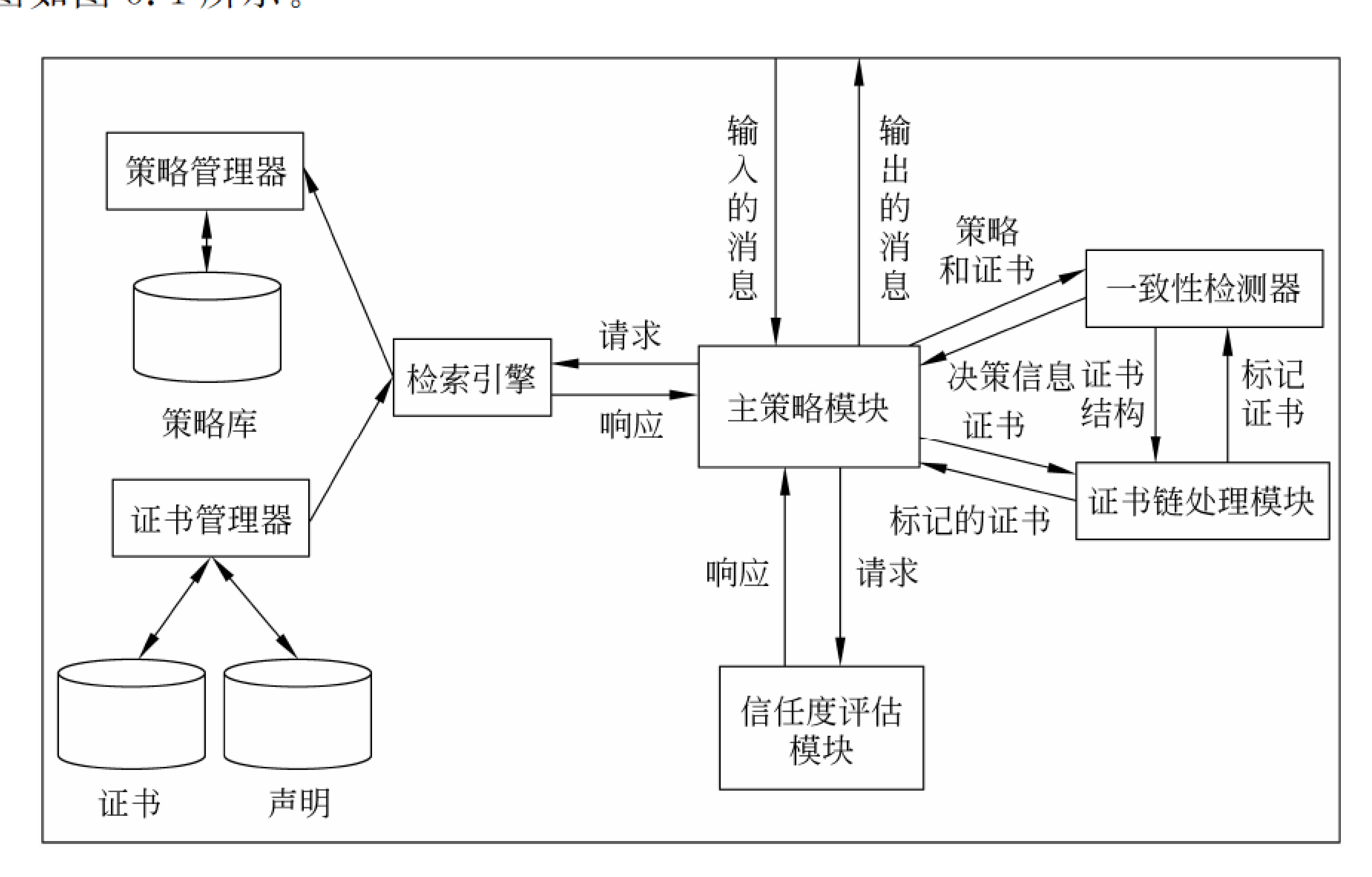


图 5.1　信任协商系统模块图

5.2　系统模块设计

5.2.1　主策略模块

当一个远程消息被可视化模块处理后，它将被传递到主策略模块。这个模块可以被称作一个协调者。对于一个信任协商会话，它将收到的消息交给合适的策略模块进行处理。当主策略模块收到一个远程消息，并且可以获取协商状态，主策略模块生成向远程协商方反

馈消息的内容。主协商模块可以无限次地调用一致性校验模块、证书链处理模块以及策略管理器、证书管理器。

主策略模块(见图 5.2)主要提供如下方法。

Nextstep():该方法的参数为协商消息,返回类型也为协商消息类型。该方法的功能是处理接收到的消息,并返回反馈给对方的消息,同时需要获取会话 ID 以及状态信息。

processPolicy():该方法的功能是查看访问控制策略是否能够被满足,如果存在这样的信任凭证集合,那么查看这些凭证是否可以被披露。如果有凭证受保护,那么查找对应的访问控制策略。

StrategyMediator
−configured : bool
+configure() : bool +nextStep() +processPolicy() +processmessage() +processCredential()

图 5.2　主策略模块

processCredential():该方法的功能是查看凭证集合中的凭证是否被保护。如果不受保护,将其添加到反馈消息中;如果被保护,则查看访问控制策略是否已经被满足。如果满足凭证可以被披露,将其添加到反馈消息中。

5.2.2　检索引擎

检索引擎主要负责检索证书和策略。此模块接受其他模块的请求,处理这些请求,反馈检索的结果。

由于检索引擎主要提供两种类型的检索:证书检索和策略检索,所以在模块中设计了父类 Query 以及它的两个子类 ProfileManager 和 PolicyManager,实现了父类的 processQuery 方法,如图 5.3 所示。这两个子类还实现了管理凭证和策略的功能,在后面的模块对此作介绍。

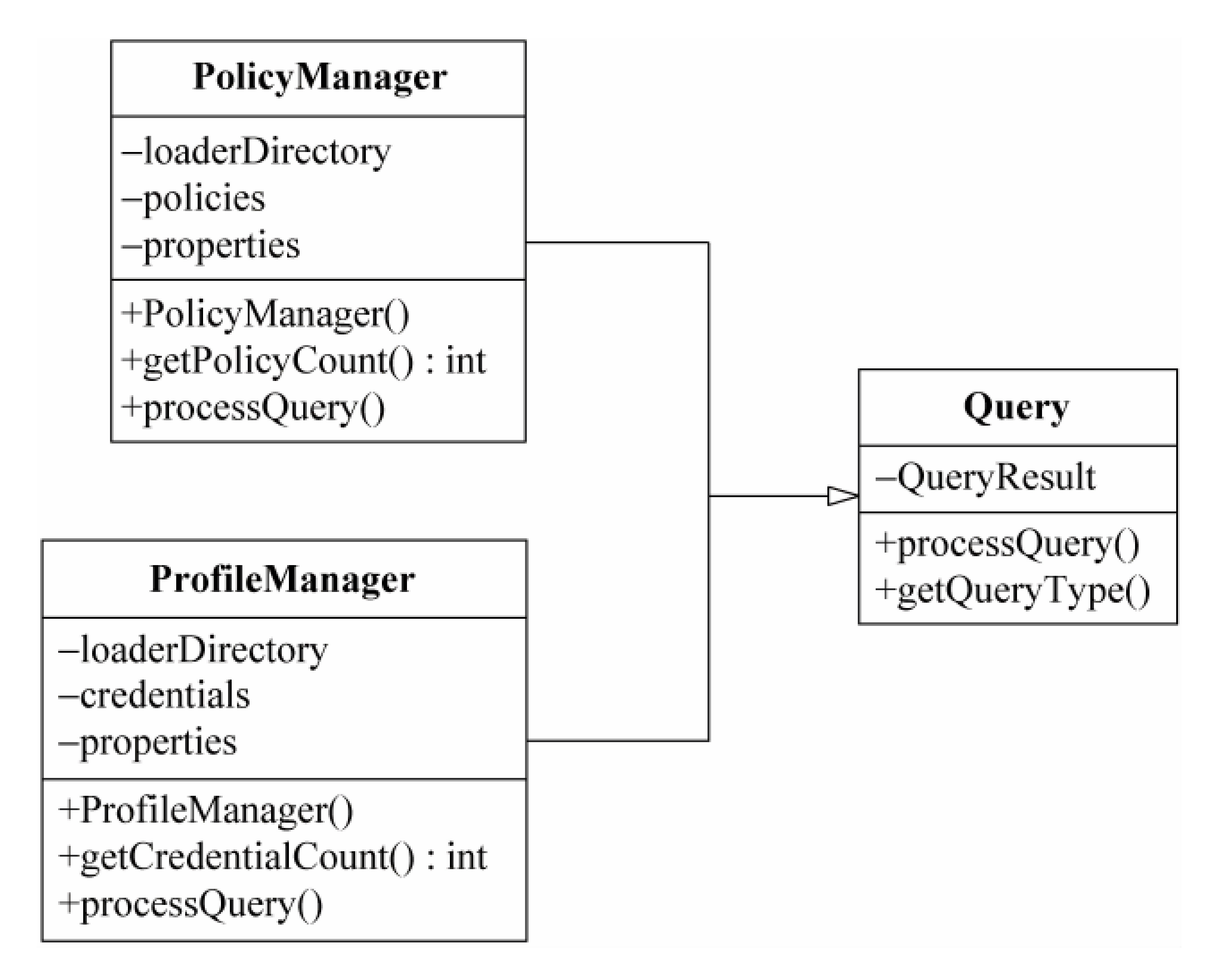


图 5.3　检索引擎

QueryResult 将检索到的证书集合或者策略集合以数组的方式返回。

检索引擎提供的方法如下。

processQuery():该方法的主要功能是获取检索类型,根据检索类型调用相应的类处理检索请求。

getQueryType():该方法的主要功能是返回检索的类型。

5.2.3 策略管理器

策略管理器负责加载用户硬盘上的策略文件,并且保证主策略模块可以通过检索引擎访问这些策略(见图 5.4)。此组件可以通过用户设置的策略文件目录来加载策略。

策略管理器提供如下方法。

getPolicyCount():该方法返回加载策略的数目。

processQuery():该方法的参数是检索请求,返回检索结果。根据请求类型,该方法可以实现检索所有的策略,也可以根据策略的 ID 检索策略。

5.2.4 证书管理器

证书管理器负责加载证书和声明,配置文件中同样需要指定证书存放的目录(见图 5.5)。

PolicyManager

−loaderDirectory
−policies
−properties

+PolicyManager()
+getPolicyCount() : int
+processQuery()

图 5.4 策略管理器

ProfileManager

−loaderDirectory
−credentials
−properties

+ProfileManager()
+getCredentialCount() : int
+processQuery()

图 5.5 证书管理器

该管理器提供如下方法。

getCredentialCount():该方法返回当前加载凭证的数目。

processQuery():该方法的参数是检索请求,返回检索结果。根据请求类型,该方法可以实现检索所有的证书,也可以检索声明。

5.2.5 一致性校验器模块

一致性校验器是自动信任协商系统的一个重要组成部分,它的主要功能是判断给定的信任凭证集合是否能够满足针对特定资源定义的访问控制策略(见图 5.6)。一致性校验器首先验证信任凭证的有效性,进行匹配时过滤掉无效的证书。

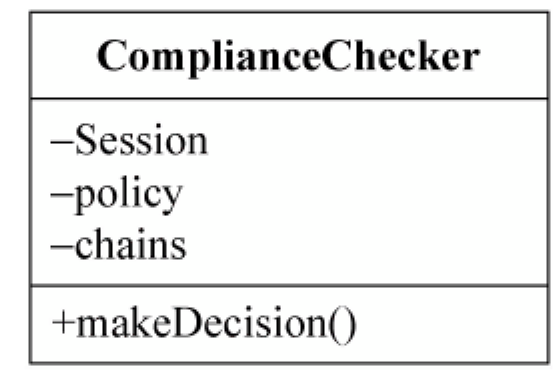


图 5.6 一致性校验器模块

一致性校验器提供了 makeDecision()方法,该方法的参数是信任凭证集合、访问控制策略和对资源的访问请求,它的返回结果是判定结果和引导信息。一致性校验器将判定的结果反馈到协商策略模块。

5.2.6 可视化模块

可视化模块的主要功能是将收到的远程消息以及发送的消息以固定的格式显示到窗口(见图 5.7)。它的输入是一条消息以及消息的方向。它分析消息的内容,并将具体内容以固定的格式显示到窗口。可以设置窗口为显示或者隐藏。

可视化模块提供了如下方法。

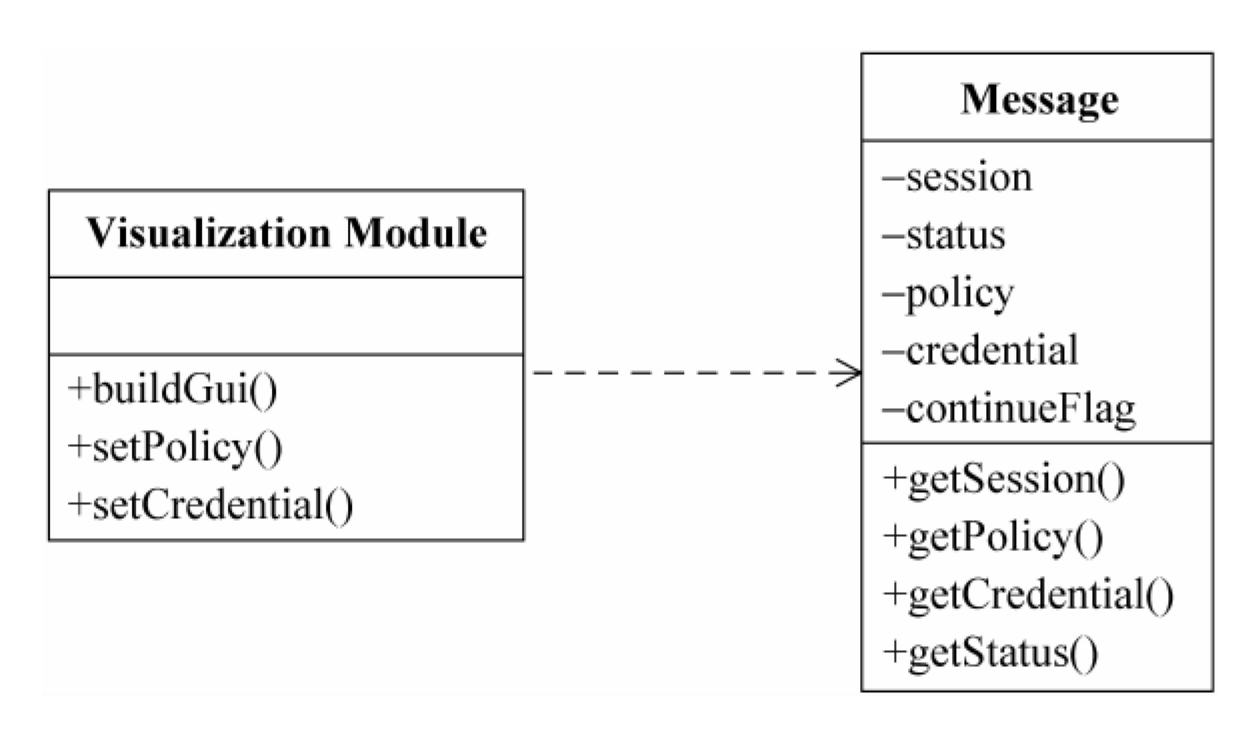


图 5.7　可视化模块

buildGui()：该方法设置输出窗口的样式以及窗口中显示的消息内容，包括消息方向、协商状态、加载消息中包括的凭证或者策略信息。

setPolicy()：该方法将消息中的策略添加到窗口。

setCredential()：该方法将消息中的证书添加到窗口。

Message 对象以固定的格式封装了消息的内容。双方之间传递的消息均被封装到 Message 中。主策略模块返回以及接收的消息都是 message 类型的。可视化模块的输入为 message 类型，通过解析 Message 中包含的信息进行显示。

5.2.7　信任度评估模块

信任度评估模块的主要功能是评估用户的信任度，并根据用户的信任度调整协商策略和访问控制策略(见图 5.8)。信任度评估的算法已经在前面作了介绍，这里主要介绍信任度评估模块如何获取用户的信任度，即信任度模块提供的调用接口。

信任度评估模块通过用户的 IP 获取用户的信任度 trustValue。

该模块提供了 getTrustValue()方法：该方法的输入是一个 IP 地址，函数查找信任度数据库，返回此 IP 地址对应的信任度等级。

TrustValueManager
–trustValue –IP
+getTrustValue() +setTrustValue()

图 5.8　信任度评估模块

5.2.8　外部接口设计

信任协商系统为外部程序提供了一个外部调用接口(见图 5.9)。外部程序只需要调用 AATN 类，并使用 AATN 类提供的一些方法来完成信任协商的功能。外部调用程序还必须为本类提供一些配置信息，来引导信任协商会话过程。信任协商过程中双方传递的消息被封装在 Message 内，AATN 处理信任协商中对方发送过来的消息，将消息进行处理后，生成反馈的消息，封装为 Message 对象发出。

AATN
–configurations
+negotiate() +prepareIncomingMessage() +prepareOutgoingMessage() +generateFailureMessage() +generateInitMessage() +processInitMessage()

图 5.9　外部接口

外部接口提供了以下方法。

Negotiate()：该方法为本接口提供的主要信任协商方法。该方法接收远程方的消息，处理消息中包含的各种元素并产生一个响应。

prepareIncomingMessage()：该方法为 Negotiate 方法提供支持，Negotiate 通过调用该方法处理收到的协商远程方消息。在本方法的实现中需要调用信任协商系统的其他模块。

prepareOutgoingMessage()：该方法为 Negotiate 方法提供支持，Negotiate 通过调用该方法处理发送到协商远程方的消息。

generateFailureMessage()：当协商过程中出现错误时，该方法生成消息，告知协商远程方终止会话。

5.3 AATN-Jess 策略语言

5.3.1 策略语言设计需求

策略语言用来描述策略，因此策略语言的语义是否清晰，描述是否详尽，决定了策略对于信任协商系统的可读性。信任协商系统对策略语言的要求主要体现在以下诸方面[1]。

(1) 定义良好的语义。一个定义良好的策略语言应该具有简单、紧凑和定义规范的语义，即使用该策略语言编写的策略，其含义与该语言的特殊应用无关。

(2) 单调性。信任协商对策略语言的单调性需求表现在：证书与策略的披露应对用户的授权产生影响；额外证书/策略的披露只能让用户获得额外资源/服务操作的权限。

(3) 证书结合。不同证书描述了特定主体不同的特征。信任协商要求策略语言具有很强的表达能力，能够使用“交”、“并”等操作将不同的证书结合起来，以满足需要提交多个证书的策略。

(4) 认证。信任协商的参与方均有多个证书，以便通过证书交换来建立信任关系。在系统运行过程中，证书提交者需证明其拥有与证书签名使用的公钥相对应的私钥信息，以确保证书的有效性。

(5) 属性值约束。通常一个证书就是一个结构化的对象，它包含关于主体属性的信息，name-value 就是属性信息的典型代表。证书可关联到某种指定的证书类型，用来简化证书规范和管理。

(6) 内部证书约束。为了更好地评估远程参与方的属性，即使参与方使用了不同的密钥，策略也应该可以表达一些约束，用来比较属于同一主体的不同证书的值。

(7) 证书链。当某一证书中的主体是证书链中下一证书的发布者时，策略语言应提供足够的描述能力来表达和约束证书链。

(8) 传递闭包。在特定的环境中，信任关系具有传递性。这要求策略语言允许策略编写者来描述信任链中的数量和类型约束。

(9) 外部函数。在协商过程中，需要一个标准的函数库来规范对诸如日期、时间和货币等的操作和比较。

(10) 本地证书变量。当处理资源的标准离线策略时，本地证书变量可使这些策略自动地与其证书关联起来，提高策略与证书的匹配效率。

(11) 检测提交者。策略编写者可以指定策略中哪些原子策略应该由访问者提交的哪些证书来满足。

(12) 敏感策略保护。敏感策略里可能包含一些个人隐私或商业机密。策略语言具有

敏感信息保护机制，以避免/防止重要信息外泄。

（13）具有互操作语言的统一形式和使用：这种需求强调了协商方法的应用能力，即在设计策略语言时，须充分考虑其是否可以在真实的环境中使用，以及是否可以集成到已有的上下文中。

5.3.2 AATN-Jess 语言特点

Jess(Java Expert System Shell)语言[2]是1995年由美国Sandia国家实验室分布式系统计算组成员Ernest J. Friedman-Hill用Java实现的一个经过扩充的CLIPS版本。它除了继承了CLIPS的特点外，还具有支持正向和逆向推理，可以在系统运行环境下直接调用Java的类库等特点。这些特点将专家系统的开发过程同功能强大的Java语言结合起来，使采用Jess语言开发的专家系统具有良好的移植性和嵌入性，可以方便地应用到网络上的不同机器中。另外，Java多线程机制使Jess可以与其他应用程序并发执行，同步机制保证了对共享数据的正确操作，通过使用不同的线程完成特定的行为，就可以很容易地实现网络上的实时交互行为。这些特点都符合P2P网络环境下信任协商系统的要求，比较适合作为P2P网络信任协商系统中的安全策略语言。但由于Jess语言是为专家系统设计开发的，其某些地方不符合信任协商安全策略语言的要求。因此本节在Jess基础上设计了AATN-Jess语言，对原有Jess语言进行了简化与提升，使其更加符合自适应信任协商系统的要求，增强了协商的效率。

AATN-Jess在保留了Jess语言的环境友好、方便用户编写策略等特点的同时，对Jess的语法结构进行了修改，取消了Jess语法结构中需要将证书中涉及的内容封装为若干对象的要求，使得AATN-Jess语法更容易理解，同时也提高了信任协商系统对策略语言的解析效率。AATN-Jess相对于其他安全策略语言具有如下特点：

（1）具有良好的系统兼容性。AATN-Jess是在Jess的基础上进行简化和提高的，其也是用Java语言进行开发的。Java语言具有跨平台特性，这样有利于AATN-Jess应用于不同的系统，这一特性也符合P2P网络的要求。同时在设计过程中将代码封装为不同的类，便于开发过程中的调用，也有助于日后根据自身需要进行修改。

（2）简洁的语法结构及较高的协商效率。AATN-Jess支持面向过程的编程方式，它提供了一些语句来控制规则后件的操作流程，如使用if…then…else和while…do语句，这样它就能很有效地利用面向过程编程的优势。AATN-Jess的这些特性使系统拥有很强的知识表示能力。同时AATN-Jess去掉了Jess在编写过程中封装对象的要求，简化了语法结构，也有助于提高协商效率。

（3）完善、友好的开发环境。AATN-Jess提供了两个交互式的、命令行的开发环境，但也可以使用文本编辑器编辑代码，然后再通过系统命令以批处理的方式载入到系统中。这样使用户更容易上手，不需要了解Java知识就可以编写自身所拥有的敏感资源和敏感证书的保护策略。

5.3.3 AATN-Jess 语法结构

AATN-Jess语言包含9个语言要素，即Template（模板）、Rule（规则）、LHand-rule（规则左键）、RHand-rule（规则右键）、Pattern（模式）、Match（配比）、Function（函数）、Solt（槽

值)和 MultiSolt(多槽值),所有访问控制策略文件都是由这 9 个语言要素嵌套组合而成的,其嵌套语法如图 5.10 所示。

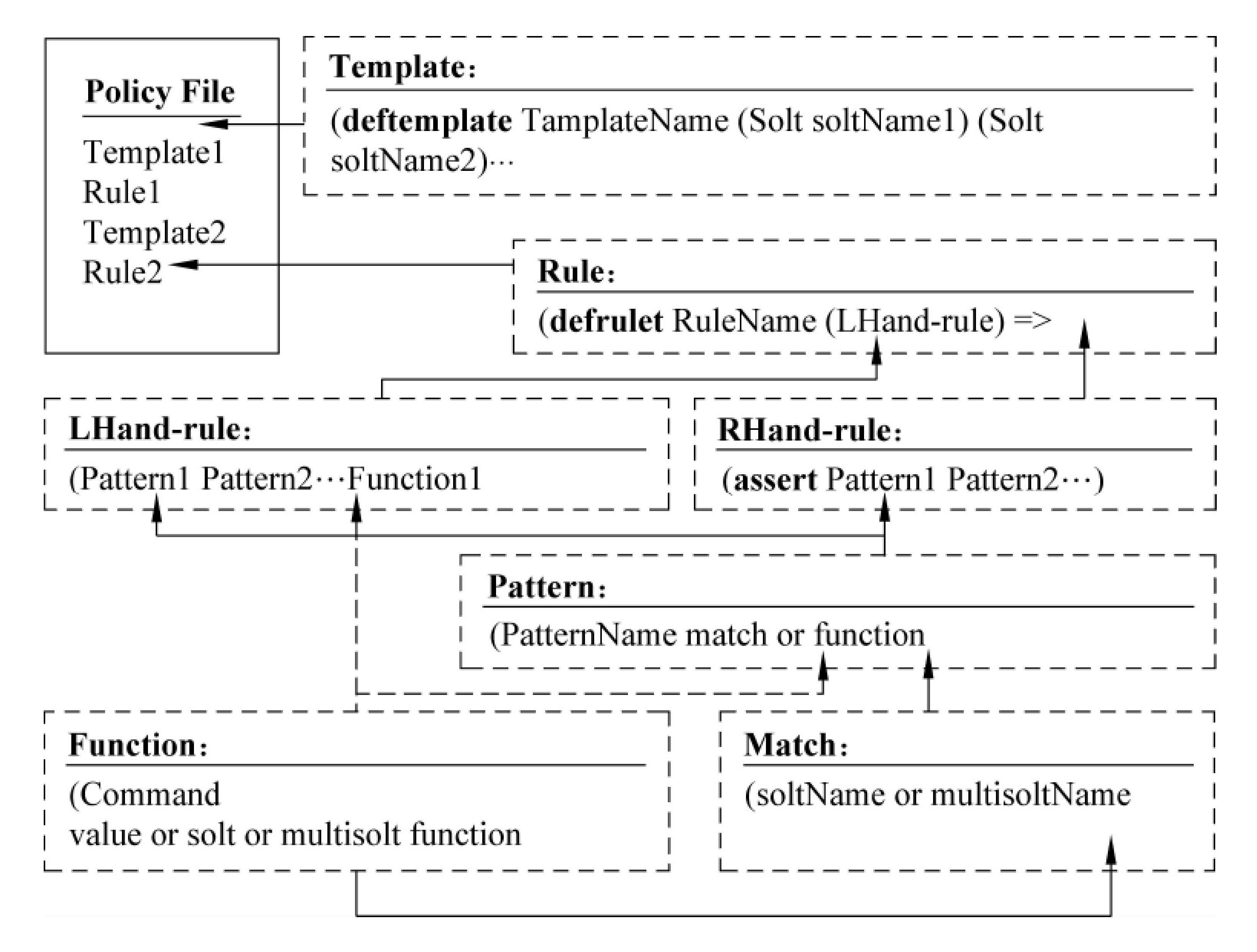


图 5.10 AATN-Jess 语法逻辑示意图

其中:

(1) 每一个访问控制策略文件(Policy File)都包含一个或者多个模板(Template)和规则(Rule)。

(2) 每一个模板是若干 Solt(槽值)和 MultiSolt(多槽值)的组合,置于一个小括号"()"内,用于存放访问控制策略要求的证书属性。

(3) 每一个规则由一个规则左键(LHand-rule)和一个规则右键(RHand-rule)组成,左键和右键之间以"=>"链接并置于一个小括号"()"内,用于定义访问控制策略对证书的约束,在证书集合能够满足规则左键的约束时,产生一个断言,即规则右键。

(4) 每一个规则左键包含若干模式(Pattern)和函数(Function),用于定义访问控制策略对证书集合的约束。

(5) 每一个规则右键包含若干模式(Pattern),用于包装规则产生的断言。

(6) 每一个模式由若干配比(Match)和函数(Function)组成。

(7) 每一个配比都包含两部分,前半部分是变量名(soltName 或者 multisoltName),后半部分是与之配比的值(value)或者约束这个值的函数(Function),意为选取该变量的值等于当前配比提供的值或者函数值的证书。

(8) 每一个函数包含一个操作符(Command)和若干操作对象(包括值、变量、值或者变量的函数)。

(9) 此外,AATN-Jess 语言涉及的数据类型和变量等遵循 Jess 语言的语法规则。

为了直观地说明 AATN-Jess 的语法,本节给出以下示例:给出一个简单的策略,以观察儿童具有的技能。有如图 5.11 所示的 Credential 1、2、3 三个证书,图 5.12 所示的策略 Policy_example.clp 可以获得所有儿童(age 属性值小于或者等于 6)的所有技能(skill)。

Credential 1:	Credential 2:	Credential 3:
issuer=O=AAAA,C=China	issuer=O=AAAA,C=China	issuer=O=AAAA,C=China
subject=O=Bob,C=China	subject=O=Alice,C=China	subject=O=Alice,C=China
attr1=age	attr1=age	attr1=age
value1=5	value1=4	value1=24
attr2=skill	attr2=skill	attr2=skill
value2=sing	value2=dance	value2=drive
...	...	...

图 5.11　标识个人信息的证书

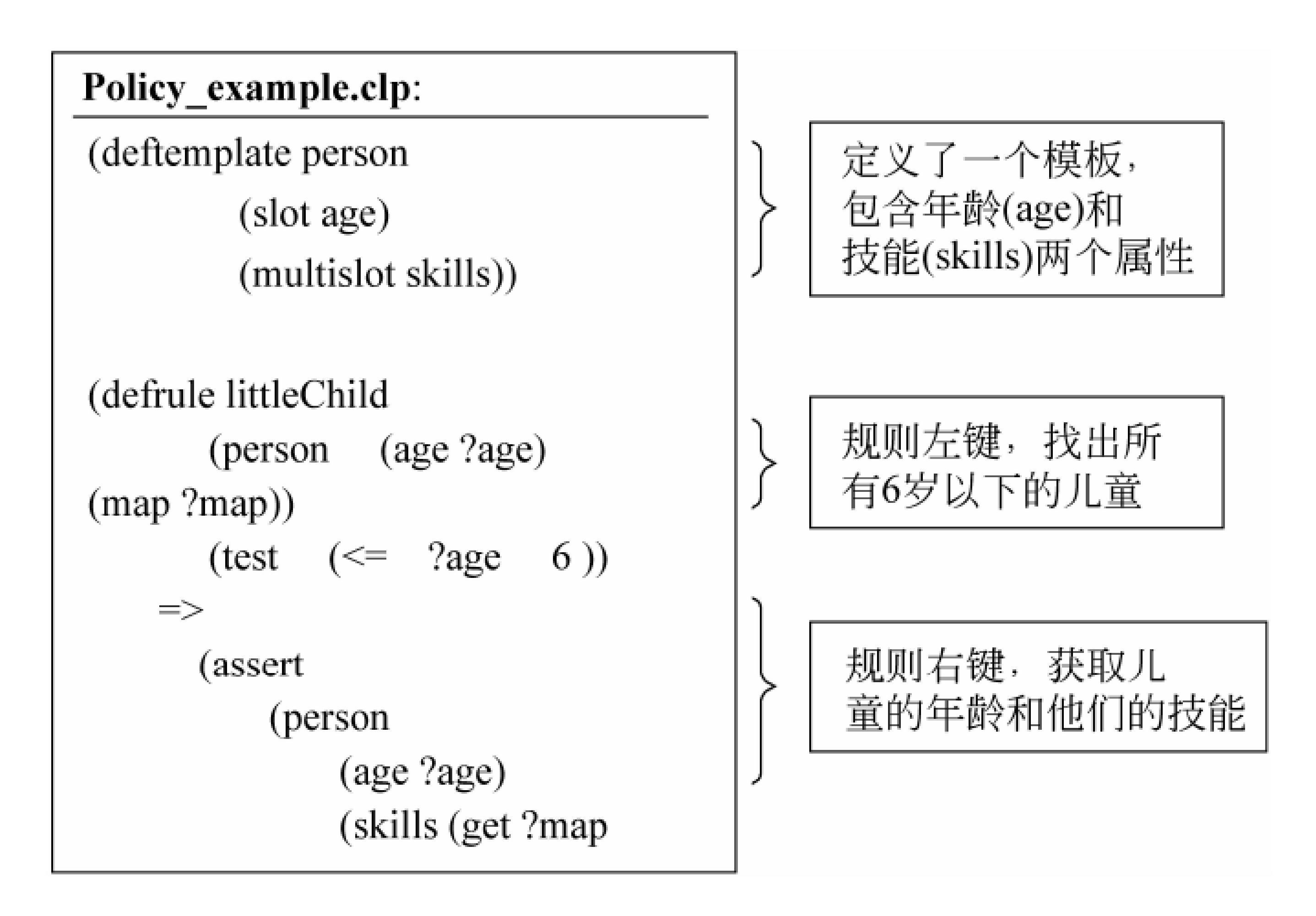


图 5.12　观察 6 岁以下儿童技能的策略

本例中，策略 Policy_example.clp 最终只命中 Credential1 和 Credential2 并获得相应的技能 sing(唱歌)和 dance(跳舞)，因为“(test (<=?age 6))”排除了所有 age 属性值大于 6 的证书。

5.3.4　AATN-Jess 策略语言编辑器

为了方便用户定义自己的访问控制策略，我们向用户提供了策略编辑器 PEditor 1.0。根据以上 AATN-Jess 语法规则，我们将 AATN-Jess 语法逻辑组织成 DOM 树(我们称之为策略编辑树)，树中每一个节点都是语法中的一个元素，用户只需在树上添加节点就可以定义访问控制策略，而不必担心语法格式错误造成的策略不一致问题。

为了演示 PEditor 1.0 的使用，我们编辑了前面示例中的 Policy_example.clp 文件，使用 PEditor 1.0 编辑的 Policy_example.clp 文件的策略编辑树如图 5.13(左边部分)所示。图 5.13 标出了策略编辑树上的节点和目标策略文件代码之间的联系，这种直观的联系便于用户理解，从而使得 PEditor 1.0 变得更加实用。策略编辑树上各节点的前置图标与各节点的节点性质名是一一对应的关系，关联如下：T—Template、S—Solt、Ms—Multisolt、R—Rule、Rl—LHand-rule、Rr—RHand-rule、P—Pattern、M—Match、F—Function，V—Value。通过不同的前置图标，用户可以了解每一个节点的意义，方便编辑和修改。

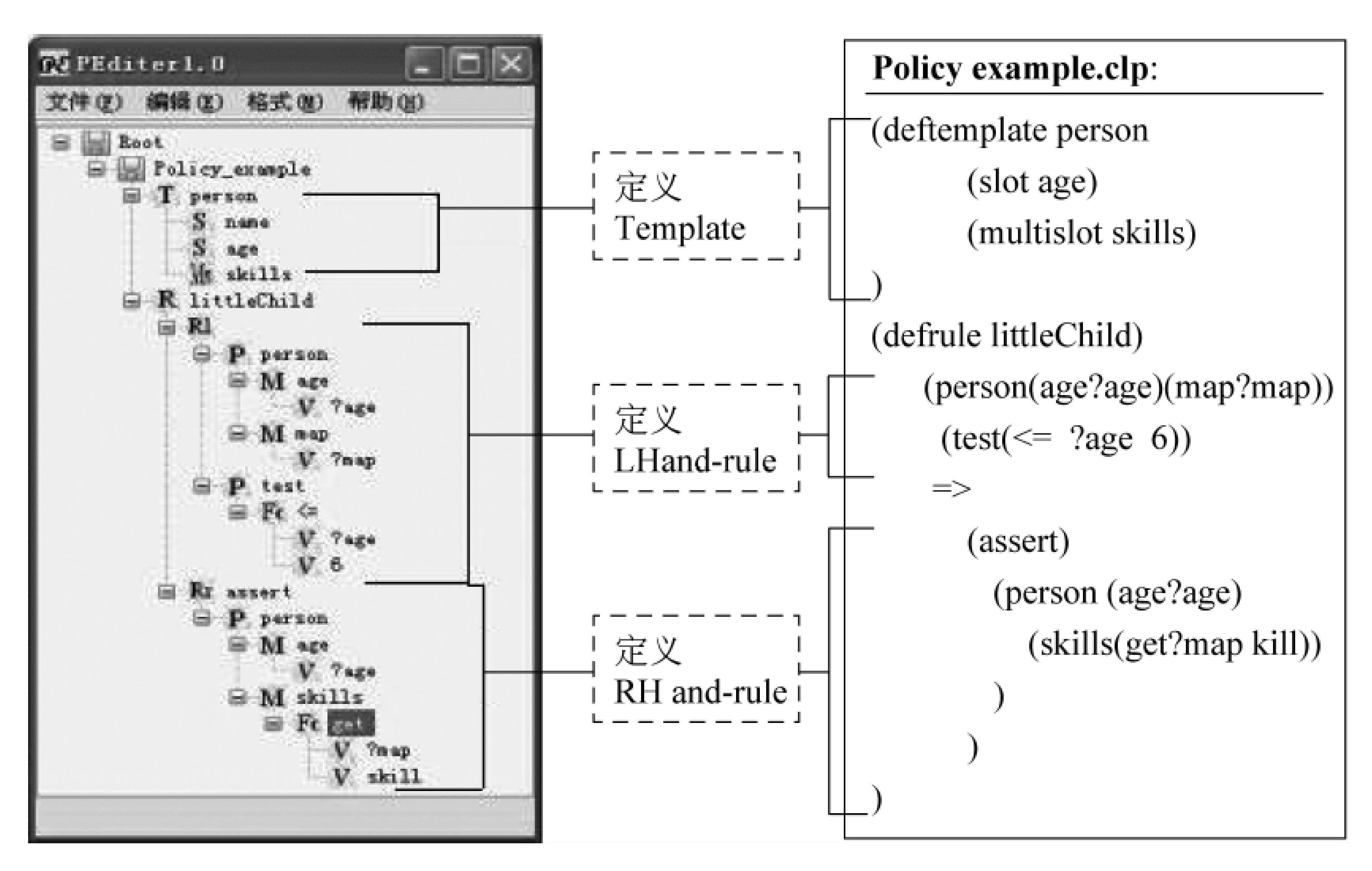


图 5.13　Policy_example.clp 策略编辑树与原策略文件的对比

5.4　本章小结

本章在第 4 章提出的自适应自动信任协商模型的基础上，主要讨论了自适应信任协商系统的设计和实现问题。首先，阐述了系统总体设计和模块设计，然后着重介绍了在系统设计和实现中使用的 AATN-Jess 策略语言，分析了 AATN-Jess 的语言特点和语法结构。

参 考 文 献

[1]　廖振松，金海，李赤松，等. 自动信任协商及其发展趋势[J]. 软件学报，2006，17(9)：1933-1948.

[2]　张国煊，张翔. 如何用开发专家系统[J]. 计算机与现代化，2003，1：29-31.

[3]　Smith B，Seamons KE，Jones MD. Responding to policies at runtime in TrustBuilder. In：Proc. of the 5th Int'l Workshop on Policies for Distributed Systems and Networks. Washington：IEEE Computer Society Press，2004. 149-158.

[4]　Shaoxu Guo，Wenbao Jiang. An Adaptive Automated Trust Negotiation Model and Algorithm [J]. In：International Conference on Communications and Intelligence Information Security. Nanning，Guangxi Province，China：IEEE Press，2010：130-134.

[5]　Wenliang Chen，Wenbao Jiang，Analysis and Design of an Adaptive Automated Trust Negotiation System，2011 IEEE International Conference on Mechatropic Science，Electric Engineering and Computer (MEC2011)，August 2011 (EI：20114114423112).

[6]　郭少旭. 自适应信任协商技术研究[D]. 北京信息科技大学硕士学位论文，2010.

第 6 章　信任管理与 P2P 网络安全

P2P(Peer-to-Peer)网络是以 Internet 物理网络为基础而构建的逻辑网络。与 C/S 模型不同,P2P 模型弱化了服务器的概念,系统中的各个节点不再区分服务器和客户端的角色关系,每个节点既可请求服务,也可提供服务,节点之间不必通过服务器就可以直接交换资源和服务。P2P 改变了 Internet 现在的以大网站为中心的状态,重返“非中心化”,并把权力交还给用户。为了共享文件,用户不需要服务器的帮助,他们之间可直接进行交互。P2P 模型还降低了对服务器的依赖并且增加了分散控制能力(相对于服务器的集中控制),没有单一的失效点。而 C/S 模型中服务器的故障或失效会使得整个系统无法正常运作。P2P 模型对等点之间通过直接连接共享资源,而且无须中心服务器的控制就能够实现对等点之间的协同合作。P2P 网络为生活中的人与人直接交流提供了网络环境,人们可以自由加入网络为他人提供内容和服务,同时也可以从网络中查找请求自身所需要的资源。随着 P2P 网络的发展,它已经成为我们日常生活、娱乐和交流的主要应用形式。目前 Internet 上基于 P2P 应用的数据流量达到 60%以上[1],这也从一个方面标志着互联网已经从集中化走向了分布化。

6.1　P2P 网络概述

最近几年,P2P 应用成为 Internet 上的一个热点。其实 P2P 并不是一个新概念,在 1969 年 Internet 的前身 ARPANET 刚出现的时候,网络的应用模式就是 P2P。ARPANET 上最早的主机包括 UCLA、SRI、UCSB 和 Utah 大学的计算机系统,这些计算机系统都是独立而且平等的,ARPANET 是以一种平等的设计方式把这些计算机系统连接起来,而不是 Master/Slave 或者是 Client/Server 的方式连接。Internet 出现后,也出现了若干经典 P2P 复杂系统,例如 Usenet 和 DNS。从 1995 年开始,随着 PC 的广泛使用并接入 Internet,使 Internet 出现了许多新情况,例如网上协作的崩溃、防火墙的大量使用,从而产生了许多非对称的网络连接方式,比如 ADSL 和 Cable Modem。Client/Server 模式成了网络主流的使用方式,严重阻碍了 P2P 网络的发展。2000 年开始,一个能够在网上进行音乐文件共享的名为 Napster 的 P2P 程序在网上广为流行,短时间就吸引了成千上万的用户,使 P2P 网络重新回到了人们的视线里。与此同时,OICQ 和 QQ 等即时聊天程序的成功也使更多开发人员关注 P2P,并开发属于他们的 P2P 程序,越来越多的用户在使用 P2P 应用程序。时至今日,P2P 技术在文件共享、即时聊天、协同工作、游戏以及网络搜索等领域均发挥着极其重要的作用。

6.1.1　P2P 网络的定义

P2P 即对等计算或对等网络,通常简称为 P2P。在 P2P 网络环境中,成千上万台彼此连接的计算机都处于对等的地位,整个网络一般来讲不依赖专用集中服务器。网络中的每

一台计算机既能充当网络的请求者，又能对其他计算机的请求做出相应，提供资源与服务。通常这些资源和服务包括信息的共享与交换、计算资源(CPU)的共享使用以及存储资源(如缓存和磁盘空间)的使用等。

对于 P2P 的定义，不同的机构有着不同的理解，每种理解方式本质上并不矛盾，都从不同的侧面揭示了 P2P 网络的特点。

Intel 公司将 P2P 定义为“通过系统间的直接交换达成计算机资源与信息共享的系统”，这些资源与服务包括信息交换、处理器时钟、缓存和磁盘空间等。

IBM 公司对 P2P 的定义则更为广泛，认为 P2P 是由若干互联协作的计算机构成的系统，系统具备以下特征[2]：

- 系统依存于边缘化(非中央式服务器)设备的主动协作，每个成员直接从其他成员而不是从服务器的参与中受益。
- 系统中成员同时扮演服务器与客户端的角色。
- 系统应用的用户能够意识到彼此的存在而构成一个虚拟或实际的群体。

从学术研究的角度看，P2P 包含 3 个层面的含义：

(1) P2P 实现技术。指构建 P2P 应用系统时所用到的技术，包括相关协议(如 Gnutella、FastTrack 等)。

(2) P2P 通信模式。P2P 通信模式与传统的客户机/服务器模式不同，每个通信方都具有相同的逻辑能力，并且每个通信方都有能力发起一个通信过程。

(3) P2P 网络。指由 P2P 节点、附属管理设备(如索引服务器等)及其相关应用等组成的可实现 P2P 功能的网络，它是一种运行在 Internet 上的动态变化的逻辑网络。每个 P2P 系统都对应一个 P2P 网络。P2P 网络是一种具有较高扩展性的分布式系统结构，对等概念是指网络中的物理节点在逻辑上具有相同的地位，而非处理能力的对等。

简单地说，P2P 技术是一种用于不同 PC 用户之间、不经过中继设备直接交换数据或服务的技术。在 P2P 网络中，每个节点的地位都是相同的，具备客户端和服务器双重特性，可以同时作为服务使用者和服务提供者。由于 P2P 技术的飞速发展，Internet 的存储模式将由目前的“内容位于中心”模式转变为“内容位于边缘”模式，改变 Internet 现在以大网站为中心的状态，重返“非中心化”。

6.1.2 P2P 结构与 C/S 结构的比较

早在 20 世纪 90 年代初期，C/S 模式就成为计算机网络中最流行的模式[3]，如图 6.1 所示。在 C/S 模式中，数据信息都保存在服务器中，如果用户想下载特定的文档，客户机需要先定位一个正确的服务器，然后向该服务器发送对文档的请求，并取得返回结果。这种模式的优点在于安全性好，易于管理，符合市场的需求。

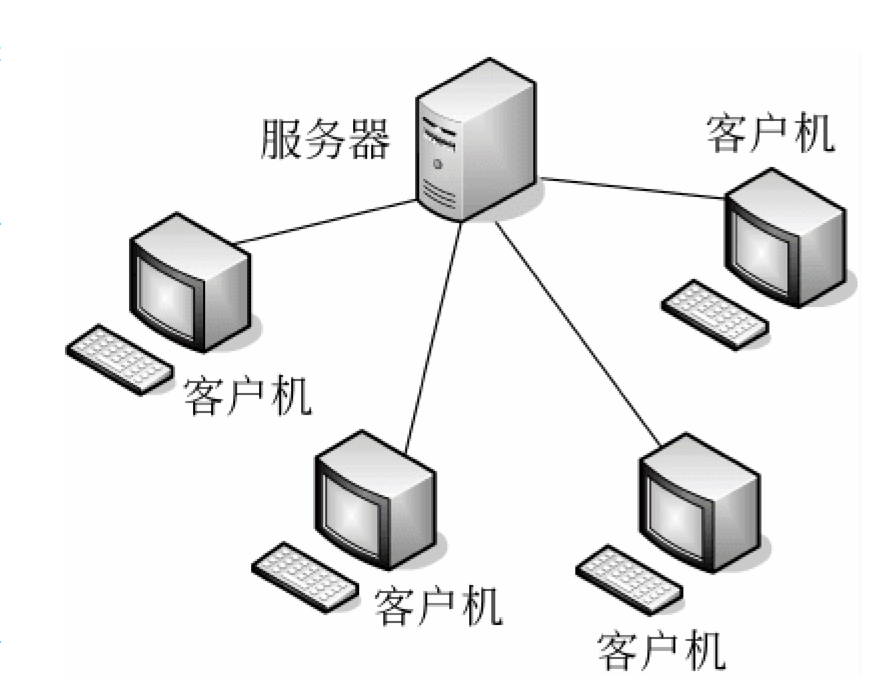


图 6.1 传统 C/S 网络结构

然而，这种模式要求有高性能的服务器，并且容易导致网络中的内容都集中于少数服务器，服务器成为网络中的主宰。由于每台服务器所能提供的信息数量受到自身存储空间的限制，并且任意时刻它所支持的

客户端访问数量既受到自身处理能力的限制，也受到服务器所在网络吞吐能力的限制，因此容易产生服务器"瓶颈"问题。

P2P 网络如图 6.2 所示，虽然 P2P 网络一些具体的结构没有完全抛弃服务器，但是服务器的功能已经被大大弱化了。各个对等节点之间是平等的主体，每个节点既是服务器又是客户端，网络资源不再集中于一处，而是分散到每个节点，使得网络重返"非中心化"，把权力交还给了用户；网络应用的核心也从中央服务器向网络边缘的终端设备扩散。

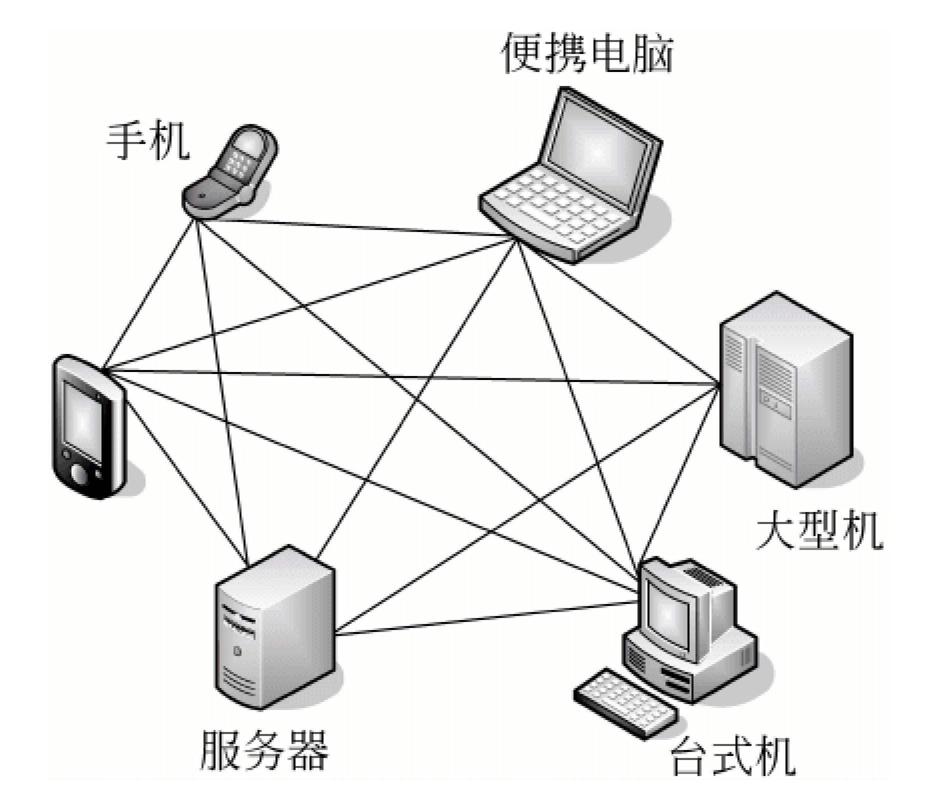


图 6.2 P2P 网络结构

P2P 网络的这种分布式结构能有效均衡负载，充分利用带宽，交互性好，及时性好，也符合市场的要求。P2P 技术是人们在 Internet 流量的主要应用类型，甚至经常超过对 Web 的访问流量。在 2002 年，对一个大型 ISP 主干网的多个边界路由器的流量分析中可知，3 个流行的 P2P 系统占用了总计达 1.2TB/天的流量[4]。

与传统网络的客户机/服务器模式截然不同的网络架构使得 P2P 网络面临着一系列的安全挑战。传统的客户机/服务器结构通过服务器对用户身份及网络中传送的数据进行了合法性验证，同时由于用户直接从服务器得到数据，因而减少了中间传送者引起的安全问题。可以说，传统的客户机/服务器模式中，服务器的存在在某些方面为网络的安全提供了保证。当然，不可忽视的是，由于服务器在网络中的特殊地位，使其成了黑客攻击的焦点，从而很容易成为安全问题的瓶颈，一旦服务器遭到攻击，则很可能导致整个网络处于瘫痪状态。传统网络与 P2P 网络的对比见表 6.1。

表 6.1 传统网络与 P2P 网络对比

对比项	传统网络	P2P 网络
服务器	有中心服务器，容易成为网络服务的瓶颈，是 DDoS 攻击的主要目标	无中心服务器或服务器功能大大弱化，不易形成网络的瓶颈
客户端	客户端必须与服务器建立连接，客户端之间无法自由通信	每个客户端之间地位平等，可自由通信
路由	由 DNS 负责域名解析，客户端之间无法自由通信	网络中的每个节点都在本地维护一份路由表，负责路由查询及存储转发
网络中的数据资源	在服务器上进行存储、备份	在网络中的各个节点上进行分布式的存储备份
身份认证	由服务器负责对客户身份进行验证	网络中各节点负责验证与自己通信的一方的身份

与传统的客户机/服务器模式相比，一方面，P2P 系统中没有中心节点，各个节点处于同等的地位，某个节点的失效并不会造成整个网络的瘫痪，因而具有更好的生存性。但是，从另一方面讲，P2P 网络缺乏一个集中的权威实体来对网络中的用户与数据进行合法性验证。而且，P2P 网络特有的数据传输的匿名性也使得其数据传输的路径不可控。这个特点很容易被某些恶意节点利用来对网络进行攻击。

6.2 P2P 网络的信任机制

在 P2P 得到广泛应用的同时，它的缺陷也逐步暴露出来。由于 P2P 网络的开放性和匿名性等特点，节点与节点之间相互不了解，很难应对一些恶意节点传播病毒、木马或质量不可靠文件的现象，同时节点所拥有的敏感资源缺乏有效保护，存在被窃取的可能。这些安全性问题严重制约了 P2P 网络的发展。

因此，如何引入一种机制解决 P2P 环境中存在的欺诈行为和恶意行为，有效地保护节点的敏感资源和敏感信息已经成为 P2P 网络安全面临的主要问题。目前仅仅是采用技术性的加密和一些简单的认证手段还不足以解决建立联系的对方是否可靠的问题。这就需要一种联系双方通过协商沟通而逐步建立信任关系的机制。协商过程中，可以将不具资格或没有权限的潜在恶意用户排除，从而有效地保证建立联系的节点的可靠性，加强 P2P 系统的安全性。因此，在 P2P 这一开放的环境中构建信任协商模型具有十分重要的意义。

6.2.1 P2P 网络安全问题

P2P 文件共享系统的迅速发展给人们在网络中共享信息带来了极大的方便，但是由于 P2P 特有的自治性、异构以及动态性等特点，P2P 文件共享系统在其快速发展的过程中碰到了很多难题，从目前的现状来看，主要有以下几个方面的问题[5~7]。

1. 占用带宽资源大

P2P 文件共享目前已经成为互联网上占用带宽资源最大的应用，2005 年欧洲主要互联网服务提供商在主干网络路由器上的一个统计表明，P2P 文件共享已经占用了主干网 70% 的带宽。这也导致了宽带网络运营商们对其又爱又恨，爱的是 P2P 文件共享为他们吸引了更多的终端用户；恨的是网络流量大幅上涨，增加网络运营成本，但是由于终端用户一般采用包月收费制度，因此他们并没有从单个用户中得到更多的收益。

2. 虚假文件、病毒等恶意信息的泛滥

由于 P2P 文件共享系统拥有相当大规模的用户，当用户进行搜索时(尤其是进行热门资源的搜索时)，必然会得到大量的搜索结果。然而这些结果并不一定都是用户需要的文件，很多是名不副实的虚假文件。P2P 无法保证用户提供的文件是完整可靠的，更无法保证用户所提供的文件信息与文件一致。一个病毒的制造者可以简单地将他的病毒伪装成一个热门文件，导致网络上病毒泛滥。

3. 简单恶意节点攻击

在 P2P 网络中，最普遍存在的恶意节点攻击方式可能是：当一个节点接收到请求节点关于某个文件的查询请求时，它便自称有匹配的文件，给此请求节点返回一个响应。当请求节点将它选为下载源进行下载时，它便提供虚假的文件，甚至散播病毒或木马，给请求节点造成严重损失。这种恶意节点攻击方式在 P2P 网络中大量存在，也是最简单、最普通的攻击方式。

4. 不诚实的反馈

在基于推荐的信任模型中，交互经验信息是共享的，这就给有恶意企图的节点提供了机

会。当它接收到某节点欲获取另一节点的可信度信息时，如果它与此节点有交互记录，但是它不是提供公正的交互信息，而是贬低(或夸大)此记录来误导请求节点(单独的此类恶意节点一般表现为诋毁曾经交易过的节点；如果是节点联合作弊，则表现为对内部成员提供夸大的评价)。虽然这种行为不会立即产生危害，但是可能导致低信任度的节点甚至恶意节点被选择作为下载源从而发生交易失败或者有害的交易。

5. 合谋欺诈

不像单独的恶意节点攻击方式，合谋欺诈是一种恶意节点联合起来形成集团，互相勾结，联合作弊的一种攻击方式。内部成员之间通过多次交易并彼此给出高的评价来抬高集团内部节点的信誉度，对外部节点则提供不可信文件及虚假的评价。此类恶意节点攻击方式危害很大，当系统中此类节点比例达到一定程度时，可能会严重扰乱系统决策，甚至使系统瘫痪。

6. 具有策略的恶意攻击

这类节点很熟悉系统“游戏”规则，开始伪装为一个好节点，提供真实文件，等骗取了较高的信誉度后，便利用此信誉欺骗请求下载文件的节点，给其提供不可信文件或者虚假的推荐，信誉度因此下降。等信誉度下降到一定程度、超出系统规定的最低门限时，该节点就变为不可信节点，但是该节点有可能改变身份重新登录网络。这类节点中有一种更狡猾的节点可能会以一定的概率提供真实文件，从而使自己信誉度时钟维持在系统规定的可信界限之内，试图不被系统觉察，以长期行骗来达到个人目的。这类节点的存在无疑加重了系统防范的负担，因为它对信任模型的攻击更具隐蔽性。

7. 管理困难、安全隐患大

P2P文件共享系统没有中心服务器，导致对系统中节点的管理非常困难。P2P网络中单个节点崇尚自由，每个节点彼此独立，既是客户机又是服务器，所以没有人知道对方有什么内容。缺乏管理的P2P网络可能会成为病毒、色情内容和非法交易的温床，甚至为恐怖分子所利用。另外，与传统的客户机/服务器结构相比，P2P网络自身的开放性和自治性使得它的安全性要差得多。一个拥有众多用户的P2P网络可能会成为黑客新的攻击目标，而且分散式结构的P2P网络有利于木马、病毒等破坏性程序的传播，这将极大地威胁P2P网络的安全。

为解决上述问题，研究人员提出了许多安全方案。在P2P网络的众多安全方案当中，信任机制非常引人注目，P2P网络引入信任机制旨在解决对等网络的一系列安全问题，研究P2P网络的信任机制具有很大的现实意义。

6.2.2 P2P信任的特点

虽然信任的定义多种多样，但这些定义所具有的特点是大同小异的。在P2P网络中，信任关系应具有如下性质。

1. 信任存在于两个实体之间

信任只有和其他实体联系起来，才有存在的意义。信任是信任者和受信者两者之间的关系。

2. 信任是主观的

信任不是一个实体的客观属性，而是其他实体对它的主观评价。不同的实体对另外的同一个实体可能有不同的评价。

3. 信任是非对称的(单向的)

实体A信任实体B，不一定实体B就信任实体A，即使A和B相互信任，它们信任对方的程度也不一定相同。因为各实体提供服务的能力和使用资源的可信度必须存在差异，同时不被信任的实体完全可以相信别的实体所享有的信誉度。

4. 信任不具有传递性

实体A信任实体B，实体B信任实体C，并不一定实体A就信任实体C。传递性只是在一定条件下满足。

5. 信任具有传播性

实体之间信任关系的变化会影响其他实体之间的信任关系。因为推荐信任关系是根据其他实体的评估而建立起来的一种信任关系。例如，当一个实体有不诚实行为时，与之有过交易的实体对其评估就会很差，这样与该实体有关的信任度都会降低，类似于现实社会中的“恶名远扬”。

6. 信任是动态变化的

实体之间的信任关系不是持久不变的，受实体行为的影响，随之动态变化。合法诚信的行为将会提高信任度，反之则降低信任度。

7. 信任是上下文相关的

不同的环境使得一个实体对另一个实体有不同的信任评价。

6.2.3 P2P信任模型的分类

目前，关于信任模型的研究十分广泛，已经提出的信任模型类型多种多样，通过分析这些类型各自的特性，对信任和信誉系统的类型做如下的分类[8]。

根据是否有几种管理信任和信誉的机制，分为集中式信任模型和分布式信任模型。集中式信任模型就是将网络中所有节点的信誉值进行统一集中存储和管理；分布式信任模型则是将节点的信任和信誉值的管理和存储任务分布到网络中的各个节点上。集中式信任模型典型的例子是在线交易系统，如eBay、淘宝等。在这些系统中，进行网上交易的买卖双方在进行一次交易后，会彼此对对方的交易行为结果进行评定；而每个节点的信誉结果值存储于中央系统，代表节点的诚信度，作为未来交易中买方选择进行交易的依据。此信誉系统中的信任值是直接发生交易后的评定值，而信誉值是以往信任值的累加，因此，信任和信誉值一致。同时，信誉值的计算式简单，便于部署实施。但对于纯P2P网络环境，如Gnutella，这类信誉系统没有实际应用的可能，因为没有中央节点来集中存储管理这些评定的信誉值信息。

根据信任值和信誉值收集的范围的不同，信任模型可分为全局信任模型和局部信任模型两类。全局信任模型的目的是要综合整个P2P网对其中的某个节点的信任评价方法[8]。该方法是将网络中的所有节点之间的相互信任评价进行迭代运算，如果能证明运算收敛，那

么最终得到的就是整个网络对其中的每个节点的信任评价。全局信任模型信任评价综合了整个P2P网络中所有Peer的意见,是比较准确的,但是忽略了信任的私人化特征,对于某个特定的节点,其他节点对它的信任值是相同的。简单的全局信任模型容易受到恶意节点联合欺诈的攻击,而复杂的全局P2P网下一种基于组群的信任模型——全局信任模型需要节点之间合作处理信任信息,计算和开销都较大[9]。局部信任模型的目的是指将整个P2P网对Peer的信任评价中的部分评价进行综合,得到局部对Peer的信任评价[10]。已有的关于P2P网络的局部信任模型大多关注于提供机制,使得节点可以根据共享信息为给定节点或资源计算局部信任值。局部信任模型的优点是计算简便,计算结果收敛快;不足则在于计算结果是局部的,所反映的节点可信度具有片面性。

根据可信对象的不同可将信任模型划分为基于角色的信任模型和基于角色拥有资源的信任模型。前者指用户节点可信,则节点拥有的资源可信,目前常常采用这种可信机制;而后者在考虑节点可信的同时还考虑节点资源的可信,进一步约束节点偶尔进行恶意行为的可能,典型的如Kaza采用的Sig2Dat。这种方法不追求节点的可信度,而是强调数据的可信度。但该方法仅针对数据共享应用(如文件共享),而且无法防止集体欺诈行为,即恶意的群体对某不真实的数据集体签名。

6.3 P2P网络信任协商系统的设计与分析

P2P网络是众多参与者按照共同兴趣组建起来的一个虚拟组织,节点之间存在着一种假定的相互信任关系。但随着P2P网络规模的扩大,P2P网络中难免会存在着一些恶意的节点,它们只需加入P2P网络中便可以轻而易举地获得其他节点的敏感资源。这种情况的出现给P2P网络带来了很大的安全威胁,阻碍了P2P网络的发展。信任协商是解决这一问题的有效方法,通过对敏感资源设置保护策略,使得资源访问方必须具备相应的资格才能访问敏感资源,大大增强了系统的安全性。

采用信任协商技术,我们研制了一个P2P网络信任协商系统——基于信任协商的网络协同攻防游戏系统NetTrust。针对目前多数网络攻防游戏系统缺少协同交流,我们在系统中专门设计了一种P2P协同聊天程序。针对聊天程序中协同交流的双方在交流过程中会涉及用户敏感信息,不利于用户自身安全和系统安全的问题,将信任协商融入P2P聊天程序中,双方在一定的信任的基础上进行协同交流,大大降低了敏感信息泄露给非授权用户的风险。

6.3.1 NetTrust系统需求分析

网络攻防游戏是信息安全教学过程中的一种有效辅助手段,但现有网络攻防游戏因缺乏协同交流而使其趣味性大打折扣。目前的协同交流工具的信任关系建立在交流双方共同的兴趣爱好的基础上,这种信任关系是脆弱的,双方在交流过程中所涉及的敏感信息的安全往往无法得到保障,给用户和系统带来了安全风险。我们根据这一问题,将基于信任协商的P2P网络协同程序融入到网络攻防游戏中。

1. 系统开发语言分析

用户在与他人进行交流的过程中往往不只局限于一人,需要与多人同时进行交流沟通,

这就需要系统开发语言支持多线程技术。同时由于P2P网络的特点,网络中的各个节点自身的系统与配置均有不同,因此要求开发出的系统应具有很强的移植性,可以满足不同系统和配置的要求。无疑在众多编程语言中,Java是比较符合这一要求的。Java是一种简单的,面向对象的、分布式的、解释型的、健壮安全的、结构中立的、可移植的、性能优异的、多线程的动态语言。Java应用编程接口为Java应用提供了一个独立于操作系统的标准接口,可分为基本部分和扩展部分。在硬件或操作系统平台上安装一个Java平台之后,Java应用程序就可以运行。现在Java平台已经嵌入了几乎所有的操作系统。这样Java程序可以只编译一次,就可以在各种系统中运行。

2. 系统功能分析

P2P网络中,每个节点既要作为服务器,又要作为客户端。服务端需要完成的功能是接受服务请求,然后针对服务或资源选择策略,根据对方的证书保护策略发送相应的证书;客户端要完成的功能是发送访问请求,根据服务方的资源保护策略发送相应的证书,并且对受保护的证书发送证书保护策略;信任协商建立成功之后建立通信过程,每个节点的服务器部分接收信息,客户部分发送信息。整个系统的流程如图6.3所示。

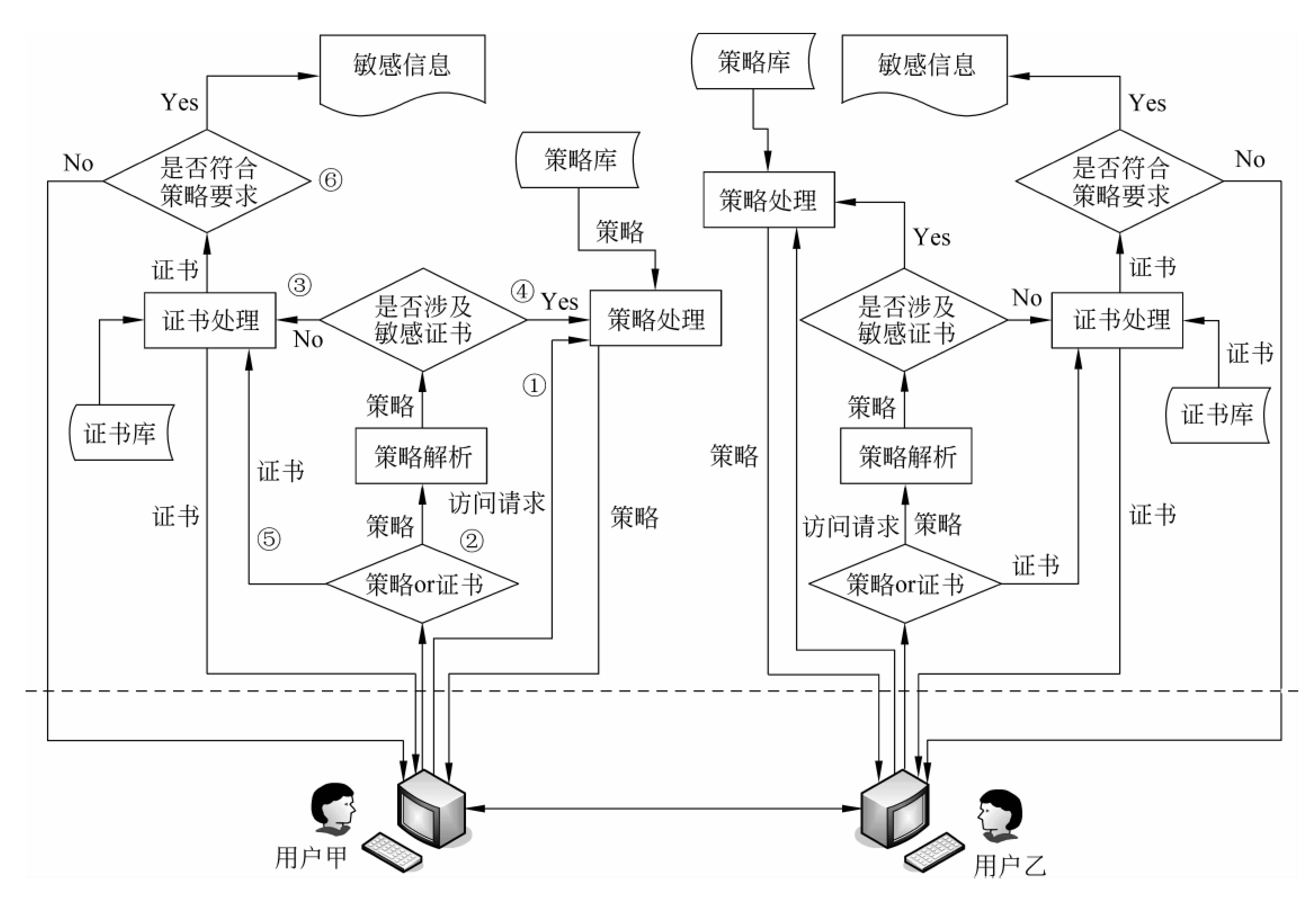


图6.3 信任协商系统流程图

其具体表述过程如下:

(1) 用户请求建立连接后,接收方通过策略处理模块为访问方提供一条访问控制策略。

(2) 用户收到返回信息后,分析它是策略还是证书,如果是策略则对其进行解析,看是否涉及自身所拥有的敏感证书。

(3) 如果收到的策略中涉及自身所拥有的敏感证书,则交由策略处理模块去处理,返回

对方关于敏感证书的保护策略。

(4) 如果不涉及敏感证书,则交由证书处理模块处理,依据对方的策略返回策略中所涉及的证书。

(5) 如果用户收到的是证书,则交由证书处理模块处理,看是否符合自身提出的策略要求。

(6) 若符合自身提出的策略要求,则双方建立信任关系,对方可同自己进行协同交流,了解自身所拥有的敏感信息;如不符合提出的策略要求,则告知对方用户协商失败,双方不能建立协同连接。

6.3.2 NetTrust 系统设计

1. 系统总体设计

NetTrust 系统的总体框架如图 6.4 所示。

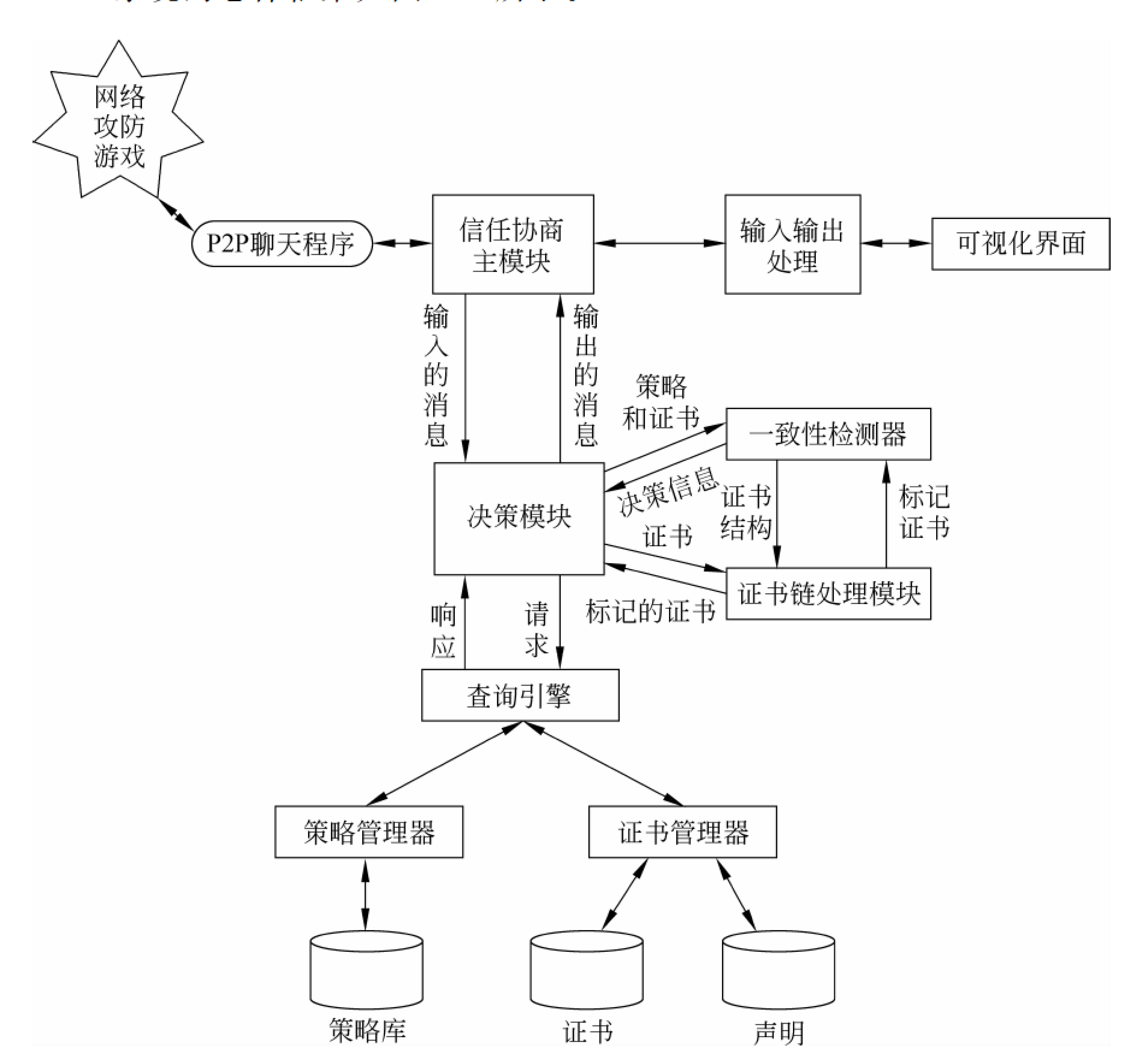


图 6.4　NetTrust 系统总体框架

各模块的功能介绍如下。

(1) P2P 聊天程序:实现点对点聊天,在点与点同意通信之前要求进行自动信任协商。

(2) 信任协商主模块:为信任协商提供了一个外部接口,P2P 聊天程序接收到的协商消息传送给该模块,处理得到的反馈消息也由该模块发出。

(3) 输入输出处理模块:接收信任协商模块传递过来的消息或者产生反馈消息,并且

将消息的某些内容传递给可视化界面。

(4) 可视化界面：主要是显示输入输出模块传送来的消息。

(5) 决策模块：协商消息经过输入输出处理模块处理后会传递给决策模块。决策模块决定下一步该怎么做，例如披露证书、传递策略、信任协商失败或成功等。

(6) 一致性检测器：主要有两个功能。一是判断给定的策略最小的证书满足集，二是对输入的证书集判断是否满足策略。

(7) 证书链处理模块：根据给定的策略找出其证书满足集的证书链；判断给定的证书链是否是所需的证书链。

(8) 查询引擎：接收查询请求，处理查询，返回查询处理结果。

(9) 策略管理器：负责从硬盘装载用户的策略库。

(10) 证书管理器：负责从硬盘装载用户的证书库和声明库。

2. 证书的设计

证书定义为包含关键字、发布者、持有人、有效日期等的属性集，证书的模板如表 6.2 所示。

表 6.2　证书模板设计

字段	id	Issuer	Subject	Datetime	map
数据类型	字符串	字符串	字符串	date	Java. hashmap

下面举个例子加以说明。证书 *C*1("ccc_food_company_certificate","北京市卫生局","乐道西餐厅","2012.04.24")是一个乐道西餐厅卫生合格证。它的发布人是“北京市卫生局”，持有人是“乐道西餐厅”，证书有效期到“2012 年 4 月 24 日”。

*C*2("c_HealthBureau_certificate","北京市卫生局","张三","2012.04.24")

证书链定义为由多个证书的关键字构成的串。由 *C*1、*C*2 组成的证书链可定义为("c_HealthBureau_certificate","ccc_food_company_certificate")。

声明的定义包含关键字、类型和值 3 个字段，具体如表 6.3 所示。

表 6.3　声明模板设计

字段	id	type	value
数据类型	字符串	字段串	字符串

例如，一个声明描述为("Bob 的电话号码","电话号码","12344678")，则该声明对象为 id="Bob 的电话号码",type="电话号码",value="字符串"。

3. 策略的设计

策略定义为 3 个字段，如表 6.4 所示。

表 6.4　策略模板设计

字段	Resource_name	claims	credentials
类型	字符串	多维	多维
描述	受策略保护的对象	声明集	证书集

例如，由只有节点拥有 $C1$、$C2$ 两个证书才可以参加某次饮食比赛(Food_Competition)的策略可描述为 $C1 \wedge C2$--->Food_Competition，用上面设计的策略可表示为 Food_Competition_policy((resource_name Food_Competition) (claims null) (credentials C)，其中 C 是由 $C1$ 和 $C2$ 构成的证书链。

6.3.3 信任协商功能的实现

在信任协商过程中，将信任协商交互双方分别称为请求方和资源或服务提供方，其中前者发起请求服务，后者作为资源或服务提供者对资源定义了访问控制策略进行保护。请求方成功访问提供方的服务或资源之前，二者要通过信任协商机制建立信任关系。下面分别介绍信任协商请求方和资源提供方的功能实现。

1. 请求方功能实现

(1) 根据本地证书类型、策略类型及决策模块等的配置参数定义一个 Trust 对象，配置参数保存在本地文件 client.properties 中。

```
final TrustBuilder2 client=new Trust (CLIENT_CONFIG);
```

其中，CLIENT_CONFIG 为 client.properties 的路径。

(2) 发送消息将请求方的配置参数告诉资源提供方，包括使用的证书类型和策略类型。请求方的配置参数通过 generateInitMessage()方法获得。同时，请求方也要知道资源提供方的配置参数，它通过 readObject()读取资源提供方发送来的配置消息。

(3) 请求方向资源提供方发送要访问的服务目标：

```
outMsg=client.generateResourceRequest(inMsg.getSessionId(),"服务目标");
output.writeObject(outMsg);
```

(4) 等待资源提供方发送响应消息：

```
inMsg=(TrustMessage)input.readObject();
```

(5) 进行信任协商循环处理过程。信任协商处理过程通过调用 TrustBuilder 的 negotiate()方法实现，negotiate()对收到的消息进行处理，将处理结果反馈给资源提供方，如果协商继续则进程挂起，等待接收下一条消息。具体代码如下：

```
while(inMsg.getContinue() && outMsg.getContinue())
{   //处理接收到的消息
    outMsg=client.negotiate(inMsg);
    output.writeObject(outMsg);
    output.flush();
    if(outMsg.getContinue()){
    inMsg=(TrustMessage)input.readObject();
    }
}
```

(6) 退出协商循环过程后，检查协商状态，修改协商标志(nego_flag，1 表示成功，0 表示失败，默认值为 0)，具体代码如下：

```
final StatusBrick status=inMsg.getStatus();
if( (status !=null) && (status.getTrustEstablished())){
    System.out.println("Negotiation success.");
    this.taMessage.append("信任协商成功了"+"\n");
    this.nego_flag=1;
    }
else{
      this.taMessage.append("信任协商失败了"+"\n");
      System.out.println("Negotiation failure.");
}
```

(7) 协商结束，关闭输入、输出流及 socket。

2. 资源提供方功能实现

(1)资源提供方的信任协商功能实现与请求方的功能实现非常相似，步骤如下：根据本地证书类型、策略类型及决策模块等的配置参数定义一个 Trust 对象，配置参数保存在本地文件 server. properties 中。

```
final Trust server=new Trust (SERVER_CONFIG);
```

其中 SERVER_CONFIG 为 server. properties 的路径。

(2) 根据协商标志 nego_flag(1 表示已经建立了信任关系，0 表示未建立信任关系，默认值为 0)判断是否要进行信任协商。

(3) 若需要进行信任协商，定义数据流，等待接收请求方有关环境配置的消息，并将资源提供方的配置告诉请求方。部分代码如下：

```
    final ObjectInputStream input = new ObjectInputStream (clientSocket.
    getInputStream());
    final ObjectOutputStream output = new ObjectOutputStream (clientSocket.
    getOutputStream());
    inMsg=(TrustMessage)input.readObject();
    outMsg=server.processInitMessage(inMsg);
    output.writeObject(outMsg); output.flush();
```

(4) 等待请求方告知想要访问的服务目标，如果未指定访问服务目标，则程序报错，协商结束。主要代码如下：

```
inMsg=(TrustMessage)input.readObject();
final NegotiationTarget target=StrategyUtils.getNegotiationTarget(inMsg);
if(target==null){
    System.err.println("No negotiation target supplied");
    return;
}
```

(5) 根据协商服务目标，进入协商处理循环过程，直到其中一方要求协商结束。主要代

码如下：

```
while(inMsg.getContinue() && outMsg.getContinue())
{     //处理从客户机发来的消息,将处理结果发给客户机
      outMsg=server.negotiate(inMsg);
      output.writeObject(outMsg);
      output.flush();
      //如果协商继续,等待客户机发送下一条消息
      if(outMsg.getContinue()){
      inMsg=(TrustMessage)input.readObject();
      }
}
```

(6) 退出循环,若协商成功,修改协商标志 nego_flag。

(7) 关闭数据流和 socket。

6.3.4 信任协商功能测试与分析

在基于信任协商的网络攻防游戏系统中,信任协商主要在 3 个部分得到具体应用。

(1) 用户登录部分：在用户登录过程中,传统的登录方式采用的是简单的用户名/密码的验证方式,一旦用户名和密码丢失,则用户信息面临着被盗的可能。将信任协商融入到用户登录过程中后,每次服务器先对用户的用户名/密码进行校验,如果符合,则服务器根据用户要访问的关卡提供不同的策略,用户根据策略提交自身所拥有的证书,服务器再验证用户提交的证书是否符合策略要求,来决定是否允许用户访问相应关卡。

(2) 游戏关卡部分：将信任协商加入到关卡中,不但能增强游戏的趣味性,同时也使用户对信任协商这一新技术有所了解。

(3) 网络协同部分：在网络协同过程中,用户之间的交流往往会泄露用户自身的敏感信息,为用户带来安全威胁。加入信任协商后,用户在建立协同关系前,先要通过信任协商对对方的身份和能力加以确认,从而建立信任关系。在一定的信任关系基础上进行协同交流,才能大幅提升游戏过程中的安全性。

1. 登录过程中的信任协商分析

传统登录方式采用对用户的用户名/密码校验的形式,这种方式方便简单,但一旦用户名和密码被木马程序窃取,用户在系统中所拥有的资源和信息就面临着被窃取的危险。融入信任协商后,用户不但要提供用户名/密码,还要根据策略提供相应的证书以证明自己的身份和能力,这增加了系统的安全性,加大了对用户的保护力度。

例 6.1 甲为基于信任协商的网络攻防游戏的合法注册用户,其在以往的过程中已经顺利通过 3 个关卡的测试。现在其重新登录游戏系统,想要开始第 4 关的攻关之旅,服务器根据甲所提交的访问的关卡数,对甲提供相应的策略,要求甲提供策略中所涉及的证书方能访问第 4 关。其具体流程如图 6.5 所示。

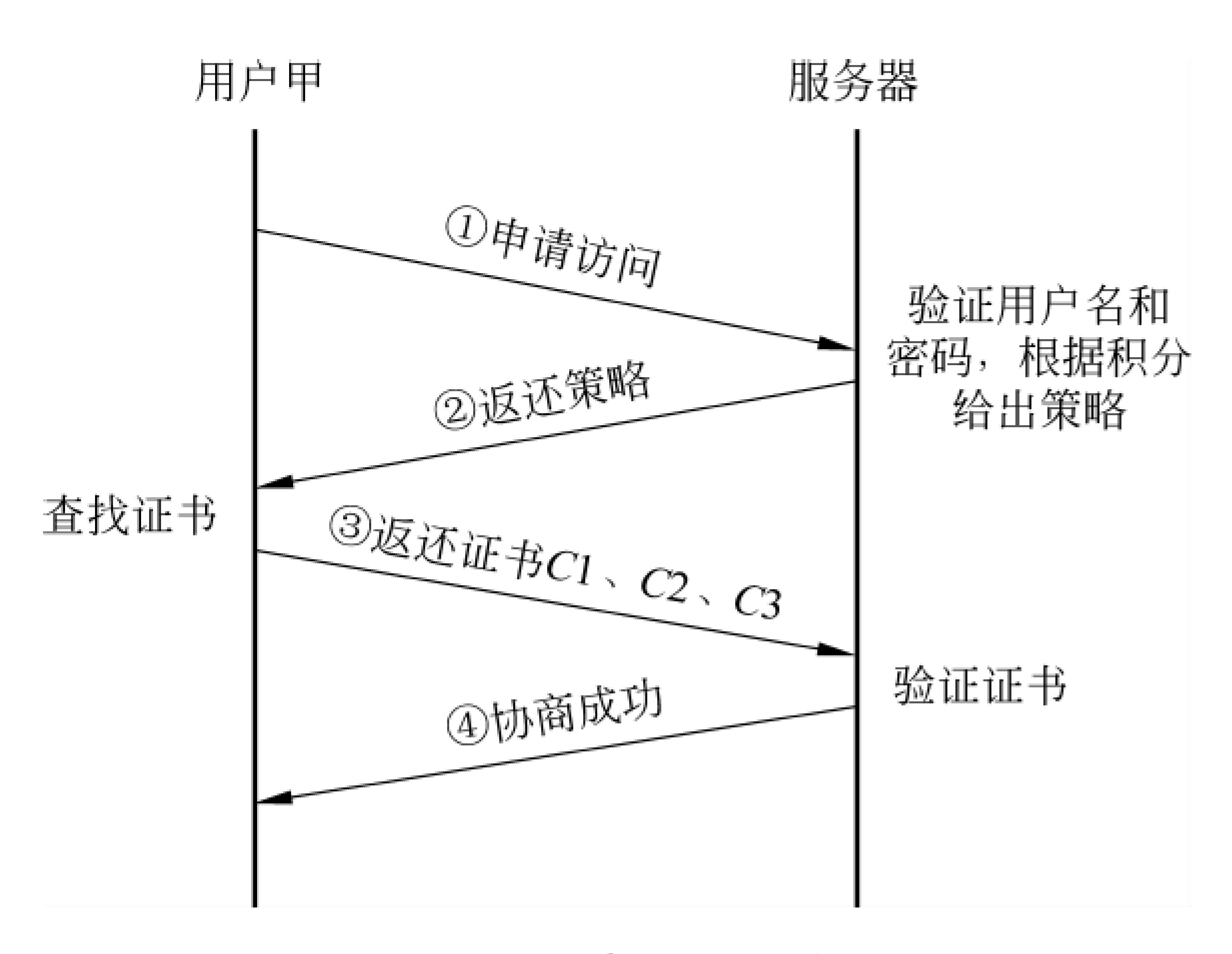


图 6.5　用户甲登录过程

甲的协商过程如下：

(1) 用户甲：提交用户名和密码，申请获得第 3 关的访问权限。

(2) 服务器：验证用户名和密码，并查询用户积分，根据积分给出甲可以访问第 3 关的策略 P1。

(3) 用户甲：搜索证书，并返回 $F1(C1 \wedge C2 \wedge C3)$。

(4) 服务器：验证用户甲传来的证书后，告知协商成功。

通过甲和服务器之间的策略和证书的交换，使服务器对甲的身份和能力有一个信任，从而允许甲访问相应关卡。如果甲不能提供服务器策略要求的证书，则双方协商失败，用户甲无法登录游戏并访问相应关卡的资源。甲与服务器协商的界面如图 6.6 所示。



图 6.6　用户和服务器的协商过程

在图 6.6 中，前两个为双方配置文件的传输，第 3 个图为用户向服务器传输所要访问的目标，第 4 个图为服务器为用户提供的策略，第 5 个图为用户根据 policy_four 的策略要求提交证书 $C1$、$C2$、$C3$。图中的按钮单击后显示的是具体的策略内容和证书内容，图 6.7 即为服务器根据用户的访问目标提供的策略。

服务器向用户 A1 提供了一条名为 four 的策略。策略要求第一个证书的 issuer 为 Server，所在地为 China，subject 为 A1，所在地也为 China，并要求第一个证书的 gate 属性为 first，并将第一个证书设为一个根节点，由此可知，第一个证书是服务器向用户 A1 颁发的通过第一关的凭证，第二个和第三个证书的要求与第一个相似。当用户提供的证书满足策略要求时，则允许用户访问资源名为 four 的关卡。在这一策略中，3 个元策略均为根节点，



图 6.7　服务器向用户提供的策略要求

下面没有子节点，且 3 个元策略之间的关系为“并”，其结构图如图 6.8 所示。

2. 协同过程中的信任协商分析

在 P2P 模式下的用户协同交流过程中，用户与用户之间往往是不熟悉的，而在协同过程中可能会涉及一些用户所拥有的敏感信息，这些信息不加以保护地透露给陌生者将会给用户的安全性带来很大的风险，用户在系统中所拥有的资源面临着被窃取的可能。因此在双方建立协同关系前通过信任协商建立信任关系可以有效地保护用户的敏感信息不被恶意用户窃取。具体在本系统中，通过对信任协商的引入，一是可以确保进行协同交流的用户身份的可信性，使系统授权的是合法用户，并具有一定的能力，从而保护协同交流中的用户信息的安全性。二是通过信任协商确定对方的身份权限，避免一些不具权限的用户参与到协同交流中，增加协同交流过程的时间开销，浪费系统资源。例 6.2 对系统中用户协同交流攻关的过程进行了说明。

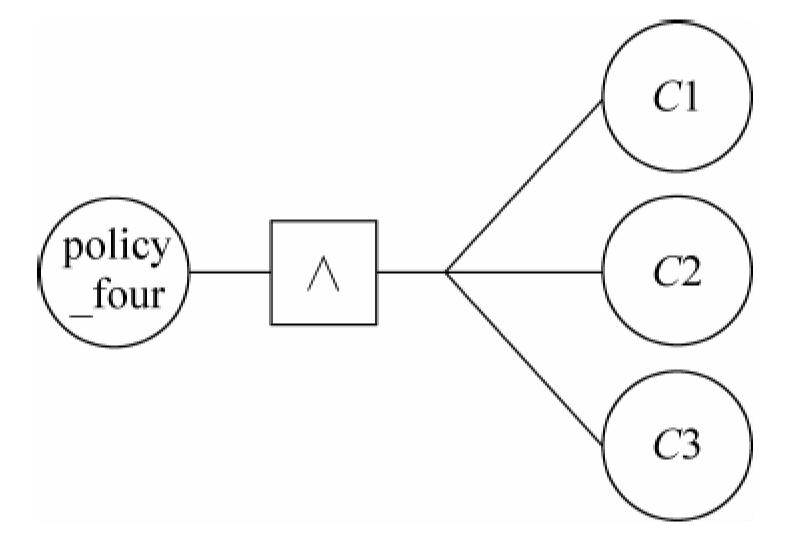


图 6.8　策略 Policy_four 的结构

例 6.2　用户甲和用户乙均到达第四关，且两人之前的积分均为满分，第四关要求两用户分别得到由系统随机生成的 1000 和 1500 以内的素数，两个素数和为通关密码。

这一过程的具体流程如图 6.9 所示。

关卡 4 的协商过程如下：

(1) 用户甲：向用户乙提出申请，请求建立协同管道，并提交自己的积分。

(2) 用户乙：根据甲的积分，返回建立沟通的策略要求 $P4 \leftarrow F4(C1 \wedge C2 \wedge C3 \wedge C4)$，并提供自身积分。

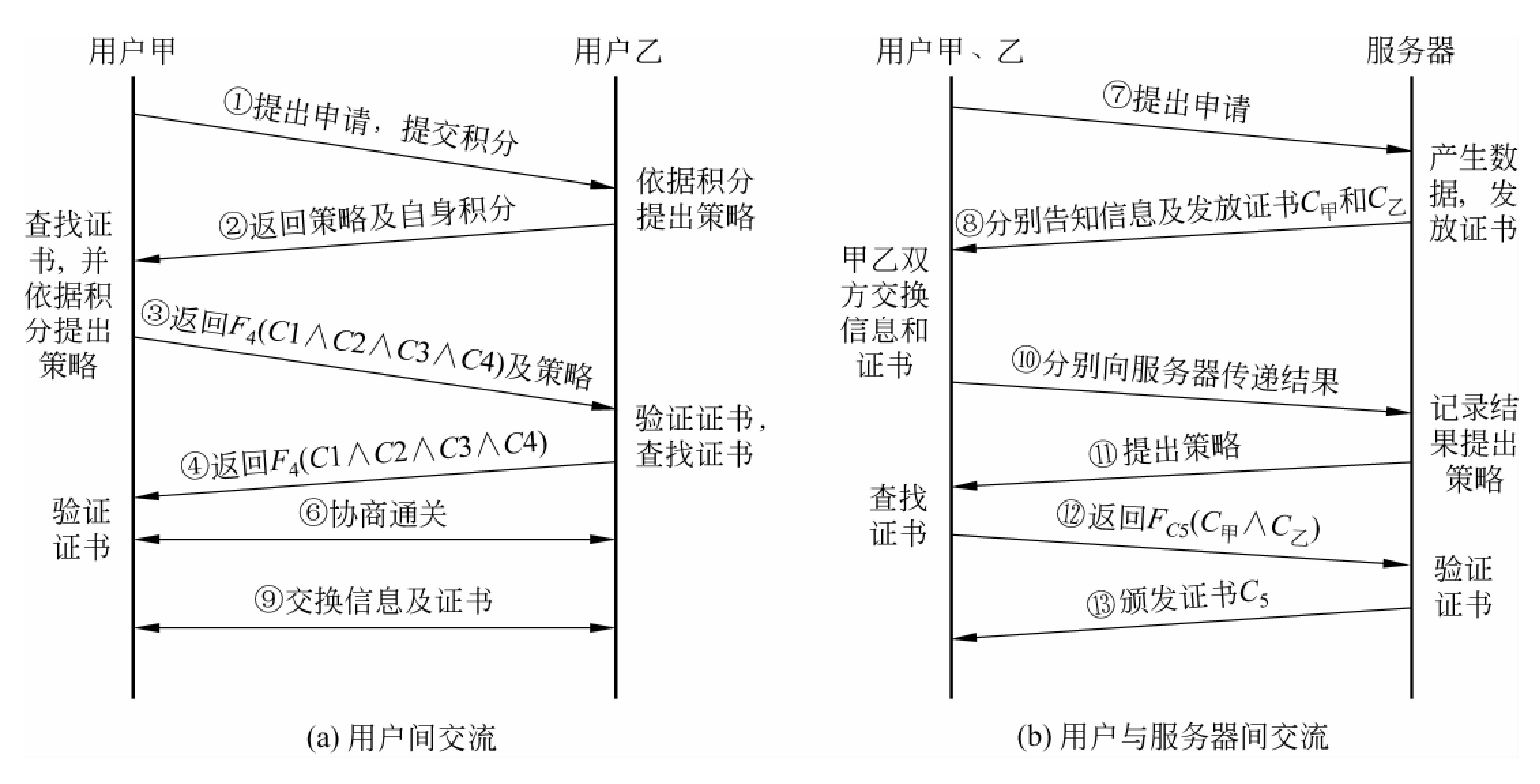


图 6.9　关卡 4 协商过程

(3) 用户甲：返回 Credential Chain($C1\wedge C2\wedge C3\wedge C4$)，并根据乙的积分提出策略 $P4\leftarrow F4(C1\wedge C2\wedge C3\wedge C4)$。

(4) 用户乙：验证甲所提交的证书，返回 Credential Chain($C1\wedge C2\wedge C3\wedge C4$)。

(5) 用户甲：验证乙所提交的证书。

(6) 双方协商成功，沟通后决定由甲获取 1000 以内的素数，由乙获取 1500 以内的素数。

(7) 甲乙双方分别向服务器发出请求。

(8) 服务器随机产生 1000 以内的素数和 1500 以内的素数，分别告知甲和乙，并为他们发放含有这两个数属性值的数字证书 $C_{甲}$ 和 $C_{乙}$。

(9) 甲和乙分别告知对方自己所拥有的数值，并交换数字证书 $C_{甲}$ 和 $C_{乙}$。

(10) 甲和乙分别向服务器提交两数和。

(11) 服务器对甲和乙分别提出访问第 5 关权限的策略 $P_{C5}\leftarrow F_{C5}(C_{甲}\wedge C_{乙})$。

(12) 甲和乙分别提交自身证书 Credential Chain($C_{甲}\wedge C_{乙}$)。

(13) 服务器验证两用户提交的证书，并根据证书的属性值的和验证双方提交的素数和是否正确。若正确，则向两人发送第 5 关的权限证书 $C5$。

(14) 双方通关成功。

这里 $C1$、$C2$、$C3$、$C4$ 和 $C5$ 分别代表 1～5 关的访问权限证书，$C_{甲}$ 和 $C_{乙}$ 分别代表由服务器发放给甲和乙的含有所选随机数属性值的证书。

协同过程中的信任协商与用户登录过程中的信任协商很相似。不同的是，用户登录过程中的信任协商强调的是用户与服务器之间的交流，而协同过程中的信任协商则是用户与用户之间的交流，其更具代表性。同时用户与用户之间的协同交流过程中可能会涉及一些敏感证书，这时可以通过策略对敏感证书进行保护。例如，在例 6.2 中，甲所拥有的关卡证书是敏感的，需要乙根据敏感证书保护策略提交证书后甲才会将敏感证书提交给乙。其在系统中的过程如图 6.10 所示。



图 6.10 协同过程中的信任协商

甲向乙提出建立协同关系的请求，这里协同关系是一个 chat_x 的抽象资源。乙根据甲访问的资源向甲提供策略，乙提出的策略中涉及的 $C1$ 证书是甲的敏感证书，甲向乙返回一个敏感证书的保护策略。乙根据该策略向甲提交证书链，即第 1～4 关的证书，甲验证后向乙提供第 1～4 关的证书链。此处的证书构造方式与例 6.1 的不同，第 1 关的证书是根节点，第 2 关和第 3 关的证书为中间节点，第 4 关为叶子节点，其结构如图 6.11 所示。

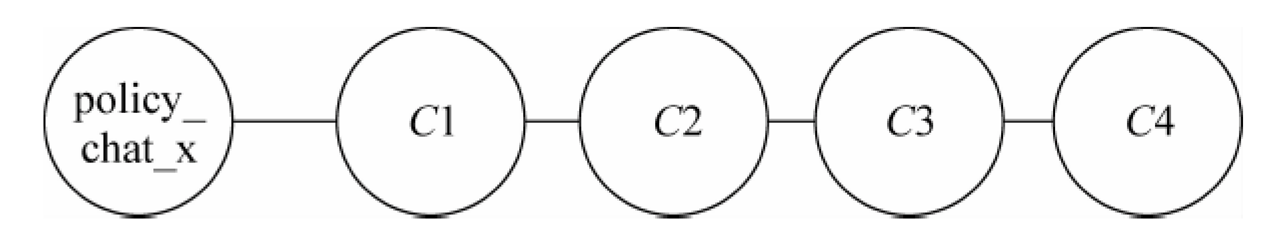


图 6.11 协同信任协商中的证书结构图

6.4 本章小结

本章主要讨论使用信任协商技术解决 P2P 网络安全问题。首先，介绍了 P2P 网络的基本概念，分析了 P2P 网络面临的安全问题以及 P2P 网络信任问题的特点。然后，着重设计并分析了一种 P2P 网络信任协商系统 NetTrust。NetTrust 是一种基于信任协商的网络协同攻防游戏系统，本章详细阐述了该系统的需求分析、系统设计和实现技术，并通过实例对系统的信任协商功能进行了测试分析，结果表明，将信任协商融入 P2P 网络聊天中，可降低敏感信息泄露给非授权用户的风险。

参 考 文 献

[1] E. Damiani, D. C. di Vimerecati, S. Paraboschi, et al. A reputation-based approach for choosing reliable resources in Peer-to-Peer networks[C]. In: Proceedings of the 9th ACM conference on Computer and communications security, Washington, 2002: 207～216.

[2] 罗杰文. Peer to Peer(P2P)综述. www.intsci.ac.cn/users/luojw/papers, 2005-11-03.

[3] 王行详，李成忠. P2P 技术安全问题研究. IECT 2005.

[4] MangWorlds，Inc. P2P：Getting Down to Business：Can Peer-to-Peer Processes Produce Profits. http://www.amazon.com/P2P-Business-Peer-Peer-Processes/dp/B00005R4KU，2001-3-4.

[5] 董西广，庄雷，常玉存. P2P 环境中的一种信任模型[J]. 微电子学与计算机，2008，25(6)：137-139.

[6] 宋超荣. P2P 网络信任模型的研究[J]. 电脑与电信，2008，1：34-36.

[7] 蒋瑜，刘嘉勇，李波. 基于信誉的 P2P 网络信任模型研究[J]. 信息与电子工程，2007，5(6)：452-456.

[8] 刘韵，卢显良，侯孟书. P2P 系统信任模型研究[J]. 福建电脑，2005，7：29-30.

[9] 窦文，王怀民，贾焰，等. 构造基于推荐的 Peer-to-Peer 环境下的 Trust 模型[J]. 软件学报，2004，15(4)：571-583.

[10] 张兴兰，聂荣. P2P 系统的一种自治信任管理模型[J]. 北京工业大学学报，2008，2：211-212.

[11] 刘思征，蒋文保，郭少旭. 一种基于信任协商的网络攻防游戏模型[J]. 北京信息科技大学学报，2009，24(4)：25-29.

[12] 刘思征. P2P 网络信任协商研究. 北京信息科技大学硕士学位论文，2009.

第 7 章　信任管理与网格安全

网格计算，作为近年来兴起的下一代网络计算技术，是解决一些大型的计算密集型或数据密集型科学计算的有效手段。与传统网络应用相比，网格计算环境具有大规模、分布、异构、动态、可扩展等特性，因此它提出了更高、更广泛的安全需求。传统的安全技术和手段，尤其是安全授权机制，如访问控制列表(ACL)和一些传统的公钥证书体系(PKI)等，不再适用于解决网格安全问题。

目前，关于网格安全解决方案的研究成果还不多，较为有影响的是 Globus Toolkit 提供的网格安全工具——GSI(Globus Security Infrastructure)。GSI 基于传统的 PKI 技术，并通过使用代理证书，在一定程度上解决了单一登录和委托等网格安全问题。然而，GSI 在授权的灵活性和可扩展性方面还有很大的不足。因此，网格安全问题的解决，还需要新的安全思想和方法。本章在全面分析网格安全需求的基础上，基于信任和信任管理的思想方法深入探讨网格安全解决方案。

7.1　网格计算概述

网格计算(Grid Computing)是近年来国际上兴起的一种重要的信息技术。它的目标是将地理上广泛分布、系统异构的各种计算资源全面整合在一起，实现网络虚拟环境下的高性能资源共享和协同工作。本节概要地介绍网格的基本概念、体系结构和关键技术。

1. 网格的基本概念

什么叫网格？简单地讲，网格是一种把地理上广泛分布的各种计算资源(各类计算机、存储系统、I/O 设备、通信系统、文件、数据库和程序等)全面整合在一起的技术，其目的是为了向用户提供相对透明的高性能计算环境，并实现计算资源、存储资源、通信资源、信息资源和知识资源的全面共享。许多人将网格计算定义为一个广域范围的“无缝的集成和协同计算环境”。

Ian Foster 是美国 Globus 网格项目的领导人之一，他这样描述网格[1]：“网格是构筑在互联网上的一组新兴技术，它将高速互联网、计算机、大型数据库、传感器和远程设备等融为一体，为科技人员和普通老百姓提供更多的资源、功能和服务。传统的互联网技术主要为人们提供电子邮件、网页浏览等通信功能，而网格的功能则更多更强，它能让人们共享计算、存储和其他资源。”

通俗地说，网格计算的基本思想，就是像人们在日常生活中从电网中获取电能一样获取高性能计算能力。几乎不会有人在打开电灯的时候考虑电是从哪个电站来的。但是，目前人们从传统的 Internet 获取信息时，必须告诉计算机去访问某一个网站，这就好比我们在打开电灯的开关时必须告诉它我们需要某一电站来的电一样笨拙。网格的目标就是让人们使用网络资源像用电一样简单。

从本质上说，网格计算需要解决的问题是如何在动态、异构的虚拟组织间实现资源共享以及协同地解决某一问题。下面对其中包含的几个概念进行解释。

1）虚拟组织

所谓虚拟组织，是由遵守资源共享规则的一组个体和机构组成，虚拟组织的典型例子有应用服务提供商、存储服务提供商、企业及企业所采用的应用所构成的系统等。虚拟组织的动态性是指组织结构、对外交互、管理模式及业务模式等是随时间变化的；虚拟组织的异构性是指各组织在目标、结构、规模、管理和运行模式等方面是不同的。

2）资源

在网格中的资源包括各类计算设备、存储设备、I/O设备、通信系统、文件、数据库、程序、信息和知识，以及天文望远镜、加速器、雷达和家用电器等仪器，并具有面向用户和透明性的特点，用户可以在不考虑资源物理位置的情况下，方便地使用资源。此外，资源也具有动态变化的特性。

3）共享

共享与以往所说的共享已有很大不同，它是在更深程度上的共享，更具目的性。它已经不再是简单的资源互连和单一使用，而是通过互连、组合和协作解决用户需要解决的问题，产生具有附加值的新服务、数据和信息等资源，满足用户的新需求。

4）协同性

协同性包括资源共享的协同性和问题解决的协同性。资源共享的协同性以资源互连为基础，既包括资源使用时不同用户因时间、空间和权限等差异引起的协商，也包括资源的组合。问题解决的协同性是指虚拟组织之间通过协作共同解决某一问题，以满足用户的新需求。

最“正统”的网格研究起源于美国政府过去十年来资助的高性能计算科研项目，以及欧洲一些团体进行的高性能计算项目。这类研究的目标是将跨地域的多台高性能计算机、大型数据库、贵重科研设备（电子显微镜、雷达阵列、粒子加速器和天文望远镜等）、通信设备、可视化设备和各种传感器等整合成一个巨大的超级计算机系统，支持科学计算和科学研究。这方面的代表性研究工作包括美国国家科学基金会资助的NPACI、国家技术网格（NTG）、分布式万亿次级计算设施（DTF），美国能源部的ASCI Grid，以及欧盟的Data Grid等。

由于网格是一种新技术，因此，目前它的精确含义和内容还没有固定，而是在不断变化，并且不同的群体可能会用不同的名词来称呼它。有人把网格看成是未来的互联网技术。国外一些媒体也经常用“下一代Internet”、“Internet 2”或“下一代Web”等词语来称呼与网格相关的技术。目前许多专家认为，网格实际上是继传统Internet和Web之后的第三次网络技术大浪潮，可以称之为第三代Internet。简单地讲，传统Internet实现了计算机硬件的连通，Web实现了网页的连通，而网格试图实现互联网上所有资源的全面连通，包括计算资源、存储资源、通信资源、软件资源、信息资源和知识资源等。

还有一类与网格相关的研究项目，其侧重点是智能信息处理，它关注的是如何消除信息孤岛和知识孤岛，实现信息资源和知识资源的智能共享，常见的名词包括语义网（semantic web）、知识管理（knowledge management）、知识本体（ontology）、智能主体（agent）、信息网格、知识网格和一体化智能信息平台等。

目前，企业界出现的许多概念和名词都与网格相关，包括内容分发（contents delivery）、

服务分发(service delivery)、电子服务(e-service)、实时企业计算(Real-Time Enterprise Computing,RTEC)、分布式计算、Peer-to-Peer Computing(简称 P2P)和 Web 服务(Web Services)等。这些名词所代表的技术有一个共同点,即将 Internet 上的资源整合成一台超级服务器,有效地提供内容服务、计算服务、存储服务和交易服务等。另一个共同点是这些技术会尽量利用现有的 Internet/Web 技术。

2. 网格的体系结构

1) 网格系统的基本构成

网格系统可以分为 3 个基本层次:资源层、中间件层和应用层。

网格资源层是构成网格系统的硬件基础,它包括各种计算资源,如超级计算机、贵重仪器、可视化设备和现有应用软件等,这些计算资源通过高速网络通信设备连接起来。网格资源层仅仅实现了计算资源在物理上的连通,但从逻辑上看,这些资源仍然是孤立的,资源共享问题仍然没有得到解决。因此,必须在网格资源层的基础上通过网格中间件层来完成广域计算资源的有效共享。

网格中间件层是指一系列工具和协议软件,其功能是屏蔽网格资源层中计算资源的分布、异构特性,向网格应用层提供透明、一致的使用接口。网格中间件层也称为网格操作系统(grid operating system),它同时需要提供用户编程接口和相应的环境,以支持网格应用的开发。

网格应用层是用户需求的具体体现。在网格操作系统的支持下,网格用户可以使用其提供的工具或环境开发各种应用系统。能否在网格系统上开发应用系统以解决各种大型计算问题是衡量网格系统优劣的关键。

2) 网格协议体系结构

在图 7.1 给出了美国著名网格计算研究项目 Globus 提出的网格协议体系结构模型。

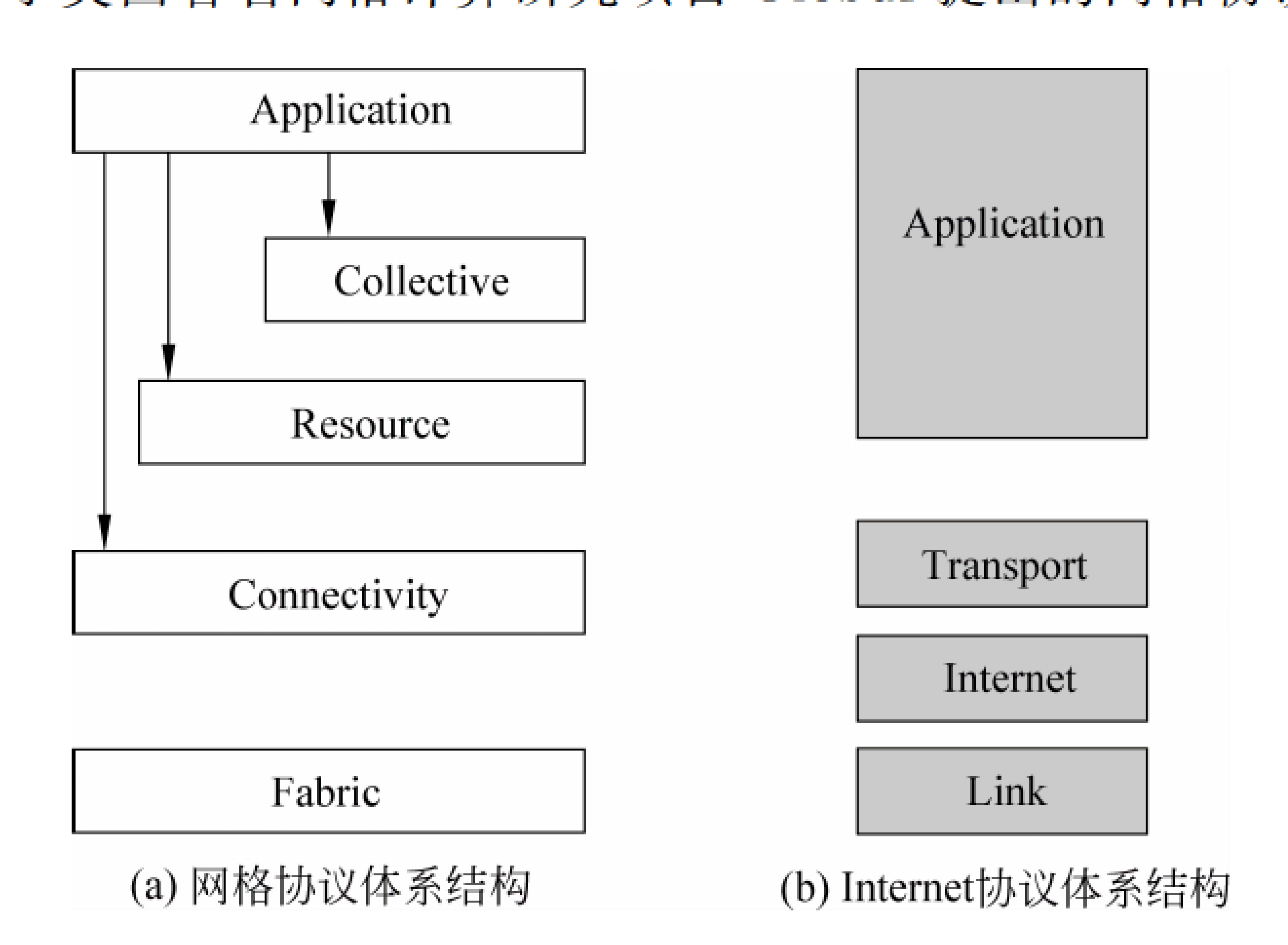


图 7.1　网格协议结构与 Internet 协议结构的关系

Globus 提出的网格协议模型充分参照了 Internet 协议模型,并以 Internet 协议中的通信、路由和名字解析等功能为基础。Globus 提出的网格协议结构分为 5 层:构造层、连接层、资源层、汇集层和应用层。每层都有自己的服务、API 和 SDK,上层协议调用下层协议的服务。

（1）构造层（Fabric）：功能是向上提供网格中可供共享的资源，它们是物理或逻辑实体。常用的资源包括处理能力、存储系统、目录、网格资源、分布式文件系统、分布式计算机池和计算机集群等。Globus 提供的网格计算工具软件包 Toolkit 中的相应组件负责检测可用的软硬件资源的特性、当前负荷和状态等信息，并将其打包供上层协议调用。

（2）连接层（Connectivity）：是网格中网络事务处理通信与授权控制的核心协议。构造层提交的各种资源间的数据交换都在这一层的控制下实现。各资源间的授权验证和安全控制也在这里实现。在 Globus 的 Toolkit 中，安全组件 GSI 提供单一登录、委托、兼容不同的本地安全方案、基于用户的信任关系等功能。资源间的数据交换通过传输、路由及名字解析实现。

（3）资源层（Resource）：作用是对单个资源实施控制，与可用资源进行安全握手，对资源做初始化，监测资源运行状况，统计与付费有关的资源使用数据。在 Globus 提供的 Toolkit 中有一系列组件用来实现资源注册、资源分配和资源监视。Toolkit 还在这一层定义了客户端的 C 语言和 Java 的 API 和 SDK。

（4）汇集层（Collective）：作用是将资源层提交的受控资源汇集在一起，供虚拟组织的应用程序共享和调用。为了对来自应用的共享进行管理和控制，汇集层提供目录服务、资源分配、日程安排、资源代理、资源监测诊断、网格启动、负荷控制和账户管理等多种功能。

（5）应用层（Application）：是网格上用户的应用程序。应用程序通过各层的 API 调用相应的服务，再通过服务调用网格上的资源来完成任务。应用程序的开发涉及大量库函数。为便于网格应用程序的开发，需要构建支持网格计算的库函数。

3．网格的关键技术

目前，网格计算的研究主要包括下面几个方面的内容。

1）网格的体系结构

从第一台计算机出现到现在，计算机体系结构已经发生了一系列变化，经历了大规模并行处理系统、共享存储型多处理器系统、群集系统等各个发展阶段，这些系统的共性是构成系统的资源相对集中。与此相反的是，组成网格系统的资源是广域分散的，不再局限于单台计算机和小规模局域网范围内。网格计算的目标是将地理上广泛分布的各种计算资源整合起来构成一台虚拟的超级计算机，因此，网格系统的体系结构是需要首先研究的问题。简言之，网格系统有哪些组成部分、组成部分之间的关系以及如何协同工作是网格体系结构研究需要解决的问题。

2）网格的操作系统

伴随着计算机体系结构的发展，计算机操作系统也经历了一系列发展变化，总的发展趋势是如何更高效、更合理地使用计算机资源。网格操作系统是网格系统资源的管理者，它所管理的是广域分布、动态、异构的资源，现有操作系统显然无法满足这一需求。

3）网格的使用模式

网格使用模式解决的是如何使用网格高性能计算环境的问题。在现有的操作系统上，计算机用户可以使用各种软件工具来完成各种任务。而在网格环境下，用户可能需要通过新的方式来利用网格系统资源。因此，在网格操作系统上设计开发各种工具和应用软件是网格使用模式研究需要解决的关键问题。

在这些研究内容中，需要解决下面一些关键技术问题。

（1）网格资源的管理。

网格环境包含各种各样的资源，这些资源具有动态变化、地域分布、系统异构等特性。在网格计算中，首先需要查清网格里所有可用资源，比如哪些主机可供访问、还空置多少处理能力、数据库里可供使用的数据是什么、共享的应用程序是否已准备好、共享主机采用何种文件系统等。资源管理的目的就是解决资源的描述、组织和管理等一系列关键问题。

（2）任务的调度与管理。

用户提交的任务要由系统来分配资源并控制其运行，包括要将其分配到哪些主机上运行、调用哪些数据、启动何种应用程序、何时开始运行等。任务调度与管理的作用就是根据当前系统负载状况，对系统内的任务进行动态调度，其调度算法及调度过程设计的好坏对系统效率的高低起着至关重要的作用。

（3）网格安全技术。

网格是通过开放的网络环境向用户提供服务的，因此它不可避免地要涉及网络安全问题。并且，与传统网络应用相比，网格的目标是实现更大范围和更深层次的资源共享，所以它存在更重要的安全问题，并提出了更高的安全需求。由于网格系统一般规模大、牵涉面广，并且拥有超强的计算能力，因此，与传统的网络入侵活动相比，如果网格系统一旦遭到攻击破坏，或者被非法利用，其潜在的损失更大，潜在的危害更严重。

与传统网络环境相比，网格计算环境极其复杂，具有大规模、分布、异构、动态、可扩展等特性，因此与传统的网络安全相比，网格安全所涉及的范围更广，解决方案也更加复杂。

（4）网格监测工具。

为了管理和维护复杂的网格环境，需要提供监视系统资源和系统运行情况的工具，即网格监测工具。网格监测工具可以监视系统的运行状态，并提供性能分析等功能。

（5）编程工具和图形用户界面。

网格系统应该能提供丰富的用户接口和编程环境。通过直观友好的用户访问接口，使用户可以在任何位置、任何平台上方便地使用系统资源。另外，网格计算的主要领域是科学计算，它往往伴随着海量的数据，面对浩如烟海的数据，想通过人工分析得出正确的判断十分困难。如果把计算结果转换成直观的图形信息，就能帮助研究人员摆脱理解数据的困难。

（6）高速网络系统。

高速网络系统是在网格计算环境中提供高性能通信的必要手段。通信能力的好坏对网格计算提供的性能影响甚大，要做到计算能力“即连即用”，必须有高质量的宽带高速网络系统支持。用户要获得延迟小、可靠的通信服务也离不开高速的网络。

7.2 网格安全需求

图 7.2 给出了网格环境下一个简单的计算示例。假设一个物理学家（P）在一个国际合作团体中进行科学研究。他收到同事的一封关于新的实验数据讨论的邮件，然后他登录到自己所在的站点 A 中的一台服务器 S1 上启动一个用户代理程序（步骤①），该用户代理程序代表他运行一个物理分析程序，而这个物理分析程序需要通过远程服务器 S2 访问存储在站点 B 中的一些数据（步骤②）。在物理分析程序运行过程中，为了比较实验结果与预期的结果，它需要启动站点 C 中服务器 S3 上的一个用于物理模拟的程序（步骤③），而该物理模

拟程序运行过程中需要访问存放在另一个站点(D)上的一些参数值(步骤④)。并且,假定物理分析程序的整个运行过程所需要的时间比较长,难以要求物理学家 P 一直守候在服务器前等待分析结果出来。

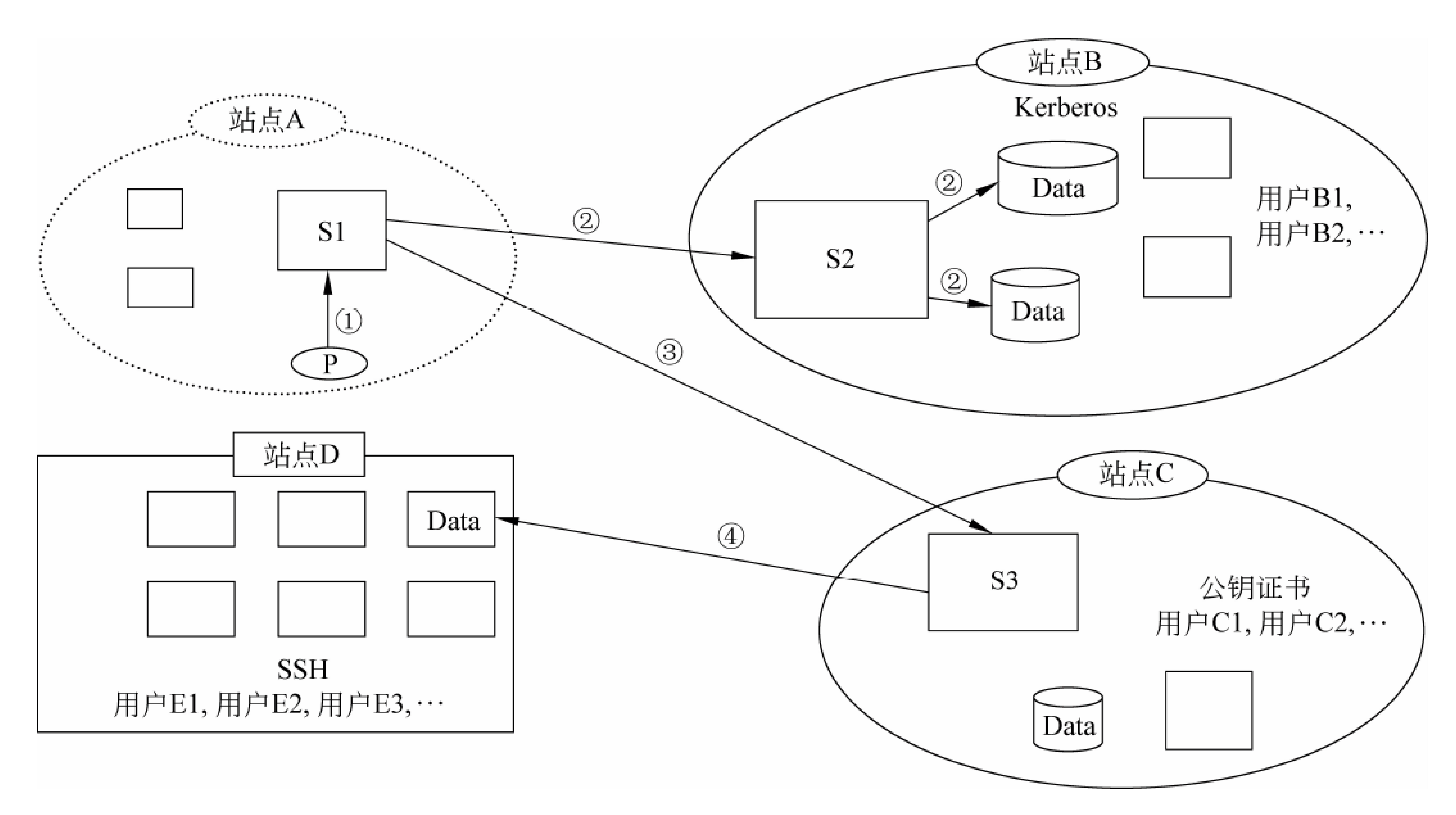


图 7.2　网格环境下的一个计算示例

上述示例虽然还不能反映网格计算的所有特征,但基本上说明了网格计算环境的以下关键特点:

(1) 资源数量巨大,并且是广泛分布、异构和动态变化的。

(2) 用户数量巨大,并且是动态变化的。

(3) 一个计算(或由计算创建的进程)可能在它的运行期间动态地要求使用或释放资源,并可能需要动态地创建许多不同的进程。

(4) 不同的资源可能要求不同的认证和授权机制。在图 7.2 的示例中,不同的站点实施了不同的本地访问控制机制。站点 B 采用了 Kerberos,站点 C 采用了公钥证书,站点 D 使用了 SSH。

(5) 网格环境下的资源和用户可能分布在不同的国家和机构里,因而可能会受到不同法律和政策的管制。

因此,与传统的网络安全问题相比,网格环境下的安全问题更加复杂,提出了更高、更广泛的安全需求,其中在认证和授权方面具有以下要求。

(1) 单一登录。用户只需在开始启动计算时进行一次"登录"(身份认证),然后就可以在无须进一步干预的情况下访问任何被授权可访问的资源,即在计算过程中获得资源、使用资源和释放资源,在内部通信时无须对用户进行再次认证。

(2) 委托。一个用户能够授予一个进程代表自己身份的权利,以便该进程可以访问用户被授权的资源。另外,如果需要的话,该进程也可以进一步委托另一个进程。

(3) 受限委托。为了减少委托凭证被破解或盗用的风险,应该能限制由委托关系继承来的权利的使用。

(4) 可兼容不同的本地安全方案。网格环境中的每个站点或资源提供者可能实施了不同的本地安全方案。整体的网格安全解决方案应该能兼容这些不同的本地解决方案,而不能要求改变本地安全方案。

(5) 基于用户的信任关系。为了便于用户在计算过程中同时使用多个资源提供者提供的资源,网格安全方案在配置安全环境时不应该要求各个资源提供者彼此之间两两交互和协商的过程。换句话说,如果一个用户有权利访问站点 A 和 B 上的资源,那么,在无须站点 A 和 B 的管理员之间进行交互和协商的情况下,这个用户就可以同时使用站点 A 和 B 上的资源。

7.3 一种基于多种证书的网格认证与授权系统

目前,关于网格安全解决方案的研究成果还不多,较有影响的是 Globus Toolkit 提供的网格安全工具——GSI(Globus Security Infrastructure)。GSI 引入了用户代理和资源代理等概念,并定义了 4 种安全操作协议(用户代理创建协议、资源分配协议、进程资源分配协议和映射注册协议)。它在 PKI 技术的基础上通过使用代理证书,在一定程度上解决了单一登录、委托等网格环境下的安全需求。但是,GSI 方案在授权的灵活性和可扩展性方面还有很大的不足,它缺乏具有高可扩展性的授权机制。在 Globus Toolkit version 1 (GT1)中提供的 GSI 方案,要求每一个访问资源的全局用户都需要在本地资源服务器上拥有一个自己的账号,每一个资源服务器都需要维护一个庞大、笨拙的全局/本地映射表,因此这种授权机制难以扩展到拥有大量资源和大量用户的大规模应用环境中。在 GT2 中提供的 GSI 方案,通过添加 CAS 功能,部分解决了网格授权的一些问题。Globus 在最近推出的 GT3 中的安全方案 GSI3,增添了一些 Web 服务的安全机制,以适应 OGSA 架构的要求。

下面在 GSI 的基础上阐述一种新的基于多种证书的网格认证与授权系统——CertGSI,它通过灵活使用标识证书、属性证书和代理证书等多种不同用途的数字证书,不但可以满足网格环境下各种基本的安全需求,而且能提供具有良好可扩展性的灵活的认证、授权及访问控制机制。

7.3.1 若干术语与定义

为了便于讨论和描述,首先约定一些相关的术语。

(1) 主体:是安全操作中的一个参与者。在网格系统中,一个主体通常是一个用户、一个代表用户的进程、一个资源(一台计算机或一个文件)或者一个代表资源的进程。

(2) 客体:是被安全策略保护的一个资源。

(3) 凭证:是用于证明主体身份的一些信息。口令和证书就是凭证的例子。

(4) 认证:是主体向请求者证明自己身份的过程,通常需要凭证进行认证。

(5) 授权:是决定是否允许一个主体访问或使用一个资源的过程。

(6) 扮演:是用于认证过程中一个主体假定为另一个主体身份的行为。

(7) 委托:是一个主体授权给另一个主体代表自己身份参与安全操作的行为。

(8) 信任域:是一个被单一管理和安全策略支配的所有主体和客体的集合。

7.3.2 CertGSI 的安全策略

利用上述术语，将 CertGSI 系统的安全策略及规则定义如下：

(1) 网格环境由多个信任域组成。该规则要求网格安全系统必须能解决大规模异构环境下的安全操作问题，能满足开放性和可伸缩性的要求。

(2) 单一信任域内的操作可以只使用本地的安全规则。网格安全策略不应该限制和影响到每个信任域本地的安全策略。

(3) 网格环境中的每个信任域都需要遵守全局安全规则。这些全局规则并不用来取代每个信任域本地的安全策略，而只是着眼于控制信任域之间的相互作用，以及域间操作与本地策略之间的映射。

(4) 位于不同信任域实体之间的操作需要进行相互认证。

(5) 所有访问控制决策都在本地做出。如果本地安全策略允许，可以根据一个主体的全局角色直接进行访问控制决策；在本地安全策略的要求下，可以将一个主体的全局角色映射为一个本地角色，然后进行访问控制决策。

(6) 一个经过认证的全局对象映射到本地对象，被假设为通过了本地的认证。

(7) 允许一个程序或进程以“用户”的身份出现，并被委托用户的部分权限。

(8) 代表同一个主体在同一个信任域内运行的所有进程可以使用相同的凭证。

7.3.3 CertGSI 的框架结构

根据上面制定的安全策略及规则，图 7.3 示意了 CertGSI 系统的框架结构。CertGSI 以多种用途的数字证书为基础，采用标识证书(Identity Certificate，IC)体现主体基本身份，使用属性证书(Attribute Certificate，AC)标明主体具有的属性，并根据主体的基本身份和属性信息作出授权和访问控制决策。它还通过使用代理证书(Proxy Certificate，IC)将一个主体的全部或部分权限授予另一个主体，实现网格环境下所要求的扮演和委托等功能，满足用户单一登录等网格环境下的特殊安全需求。

图 7.3 给出了典型的网格环境下用户进行一次计算可能涉及的各种安全因素。首先，一个用户登录到网格环境中的一台计算机，通过出示自己的标识证书及私钥、相关的属性证书，由计算机上的代理生成器创建一个用户代理，该用户代理持有一个代理证书及私钥；其次，由用户代理代表用户角色与计算中需要访问的资源站点进行交互。

在一次计算中，用户代理可能需要访问多个站点上的资源，并且在资源访问过程中可能需要启动许多进程，这些进程都需要在一定程度上代表用户角色参与相关的操作，它们持有相应的代理证书及私钥。因此，图中的用户、用户代理和进程等都属于 CertGSI 中的主体，而系统中的客体是指网格环境中通常所见的许多资源，例如计算机、网络设备、存储设备和显示设备等。

在 CertGSI 系统中我们引入了一个代表资源与用户交互的实体——资源代理(Resource Proxy，RP)。RP 持有相应的标识证书及私钥，它代表资源与访问请求者(远程主体)进行交互。首先，它需要认证远程主体的身份，并负责建立一个安全的通信通道；其次，调用策略引擎(Policy Engine，PE)根据本地的访问控制策略作出授权决策。关于策略引擎及其授权机制，将在第 7.3.6 节介绍。

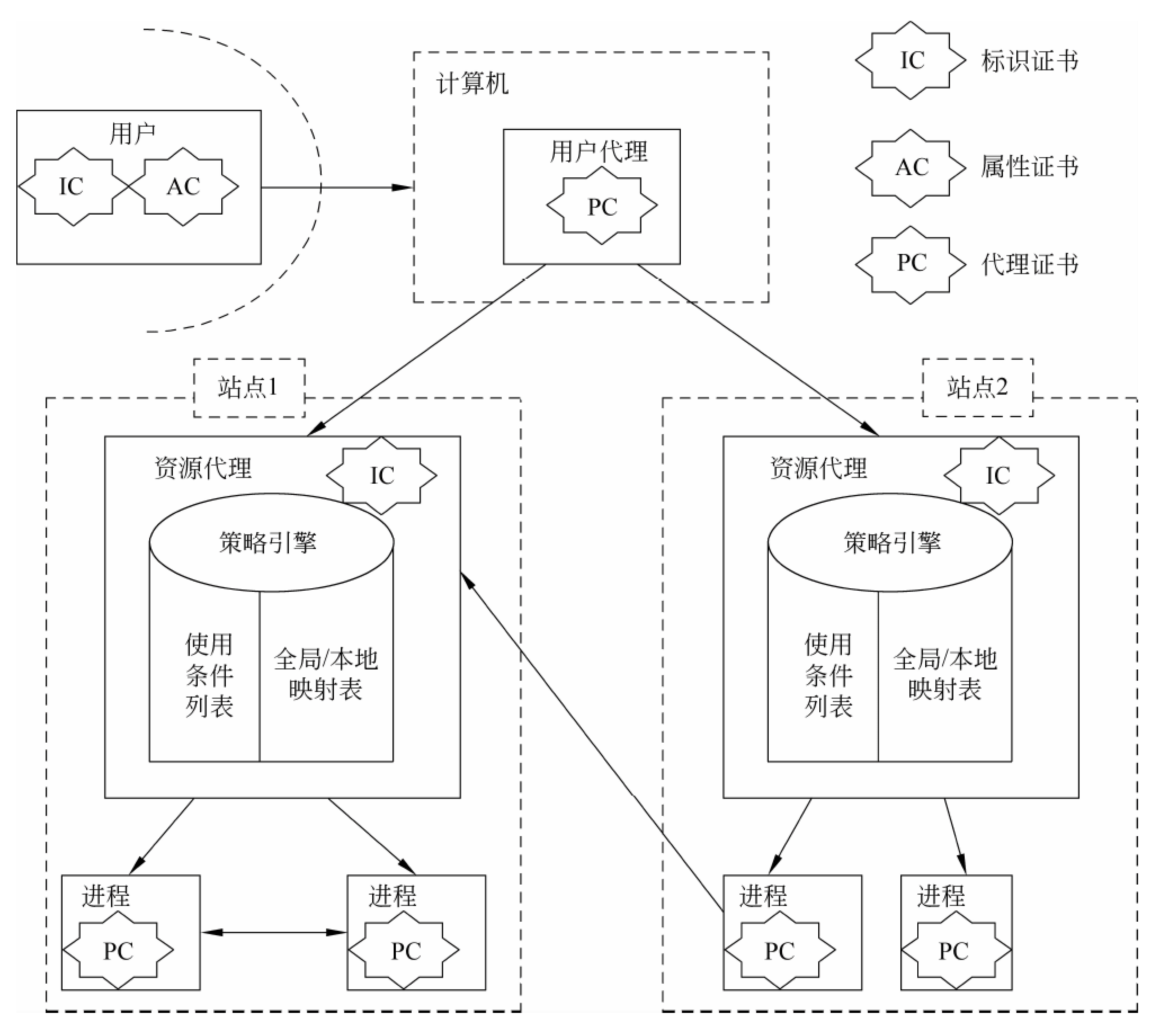


图 7.3　CertGSI 的框架结构

7.3.4　多种证书

CertGSI 共使用了标识证书(IC)、属性证书(AC)和代理证书(PC)3 种类型的证书。其中，标识证书用于标识主体的基本身份，它包含主体名称、主体公钥、有效期和颁发者数字签名等信息。主体之间交互时，每个主体以持有与标识证书中的主体公钥相对应的私钥信息为依据，证明自己的身份。标识证书由标识证书颁发机构(Certificate Authority，CA)颁发，CA 是权威可信的第三方，任何一个信任 CA 的通信一方都可以通过验证数字证书上的 CA 数字签名而获得对证书的信任，并通过验证对方拥有与证书中公钥所对应的私钥而建立与对方的信任关系。

属性证书用于标明主体具有的属性，它包含属性所有者名称、属性类型及属性值、有效期、颁发者名称及其数字签名等信息。与标识证书相比，属性证书不包含有实体的公钥，它的生命周期较短，可以灵活地根据主体某种属性的变化而改变。属性是一个主体的特殊性质，用于指明所有者的成员资格、角色、安全许可或者其他一些授权信息。虽然一个主体的属性信息也可以放在它的标识证书的扩展项中，但是这种方法存在许多局限性[2]。CertGSI 采用标识证书体现主体的基本身份，使用属性证书标明主体具有的属性，并主要依据主体的属性信息作出访问控制决策，这种机制可以大大提高访问控制的灵活性与安全性。

属性证书由属性证书颁发者(Attribute Authority,AA)颁发,AA是某种属性的管理者和授权者,它也应该持有一个由CA颁发的标识证书,拥有相应的公私钥对,并用自己的私钥对颁发的属性证书进行数字签名。

代理证书是一种临时证书,主要用于网格环境下实现扮演和委托等功能,将一个主体的全部或部分权限授予另一个主体,以满足用户单一登录等安全需求。在CertGSI中,代理证书包含了被代理用户的标识证书和属性证书中的有关信息。具体包括代理主体名称、代理主体公钥、代理约束信息、属性证书信息、有效期、颁发者名称及其数字签名等信息。代理证书由代理证书颁发者(Proxy Authority,PA)颁发,PA是被代理的用户或该用户的上一个代理。因为代理证书中包含了代理所具有的基本身份、属性及属性值等信息,所以资源的访问控制决策者根据代理证书就可以作出授权决策。

代理证书与标识证书类似,也是一种X.509公钥证书,不过它还具有以下特点:

(1) 代理证书颁发者可以是一个标识证书,也可以是另一个代理证书。

(2) 代理证书只可以用于签发另一个代理证书,而不可以用来签发其他证书。

(3) 代理证书拥有自己特定的公钥/私钥对。

(4) 代理证书只能继承标识证书的身份,而没有自己特定的身份。

(5) 代理证书可以继承颁发者属性证书所表明的属性。

7.3.5 身份认证

CertGSI是基于标识证书提供身份认证功能的,图7.4给出了基本的身份认证模型。

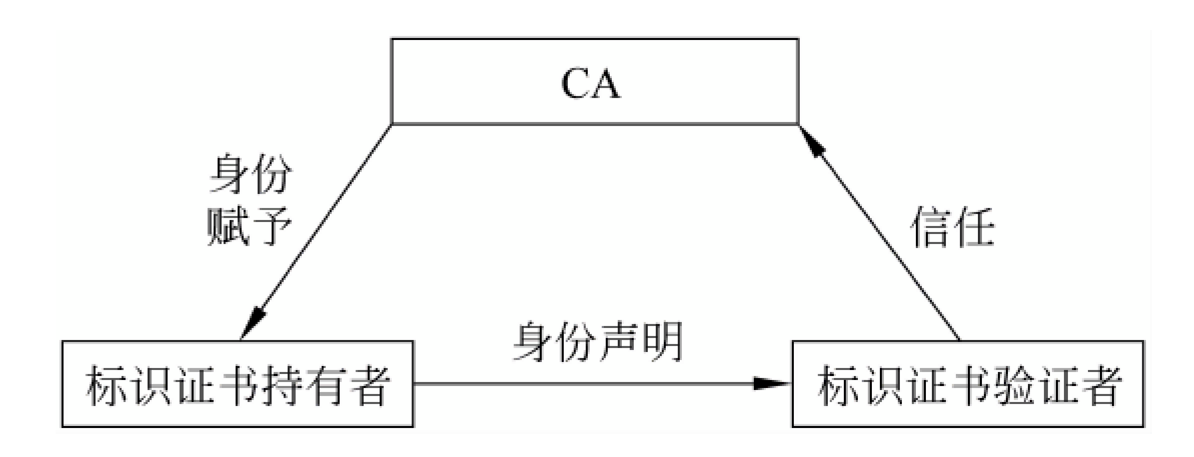


图7.4 CertGSI基本身份认证模型

在委托情况下,CertGSI主要是基于代理证书提供身份认证,图7.5给出了委托身份认证模型。

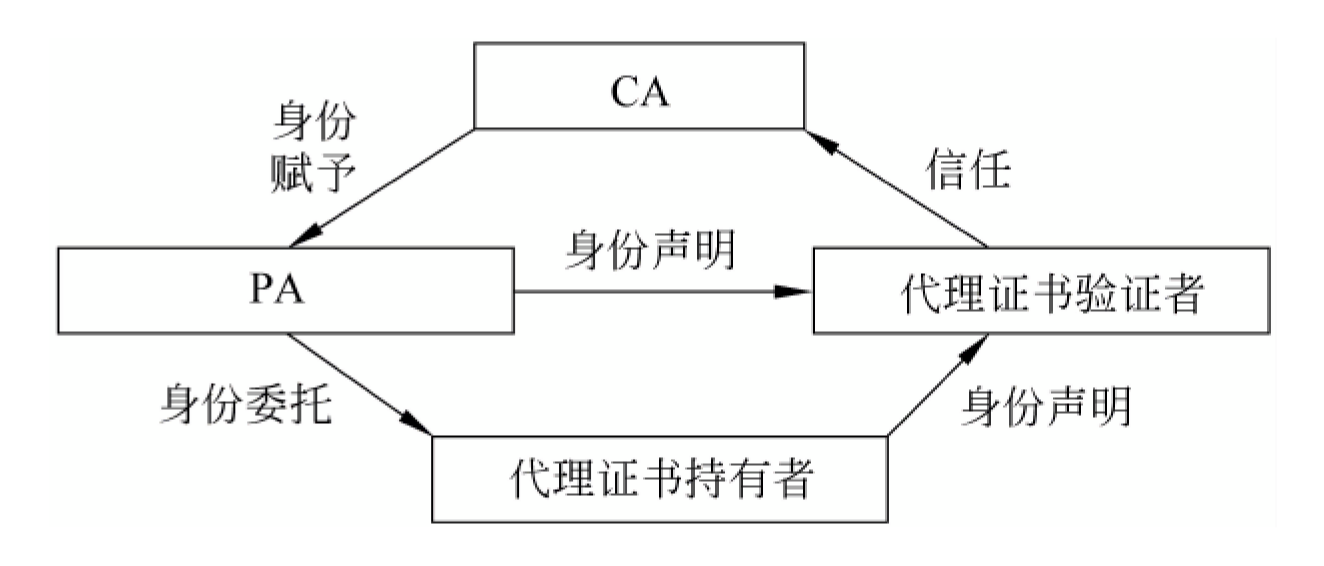


图7.5 CertGSI委托身份认证模型

CertGSI中的两个实体A和B进行双向身份认证的过程描述如下:

(1) A→B: A, N_A, C_A

(2) A→B: $PK_A\{SK_B\{A, N_A, B, N_B, K\}\}$, $K\{N_A, N_B\}$, C_B

(3) $K\{N_B\}$

其中,$PK_X\{\}$表示使用实体 X 的公钥对{}中的数据进行加密。

$SK_X\{\}$表示使用实体 X 的私钥对{}中的数据进行加密。

$K\{\}$表示使用临时产生的会话密钥对{}中的数据进行加密。

C_X 表示实体 X 的标识证书或代理证书。

N_X 表示实体 X 产生的随机数。

7.3.6 访问控制

CertGSI 授权主要是依据主体属性证书中的属性信息进行的,基本的授权模型,如图 7.6 所示。

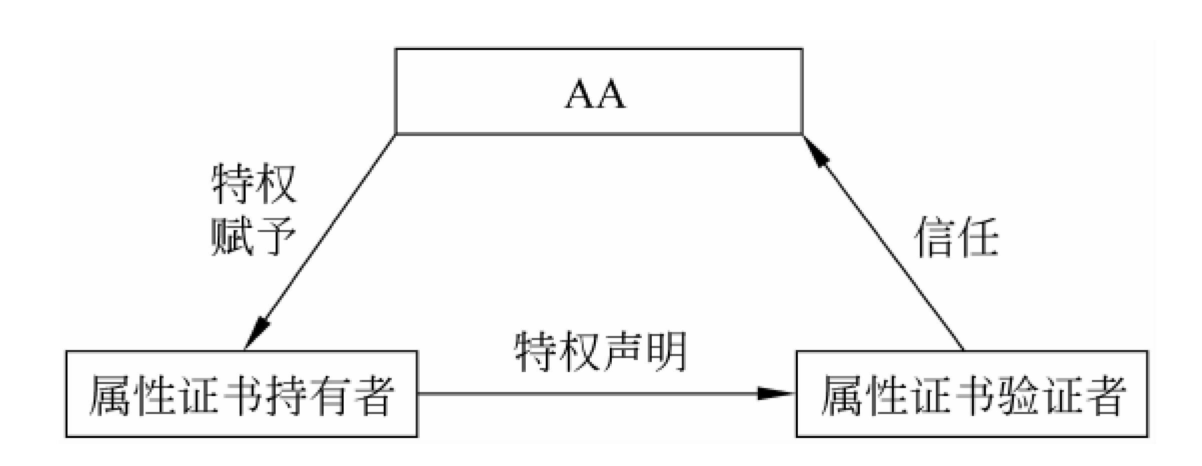


图 7.6 CertGSI 基本授权模型

在委托情况下,系统则根据代理证书承载的属性信息进行访问控制,图 7.7 给出了委托授权模型。

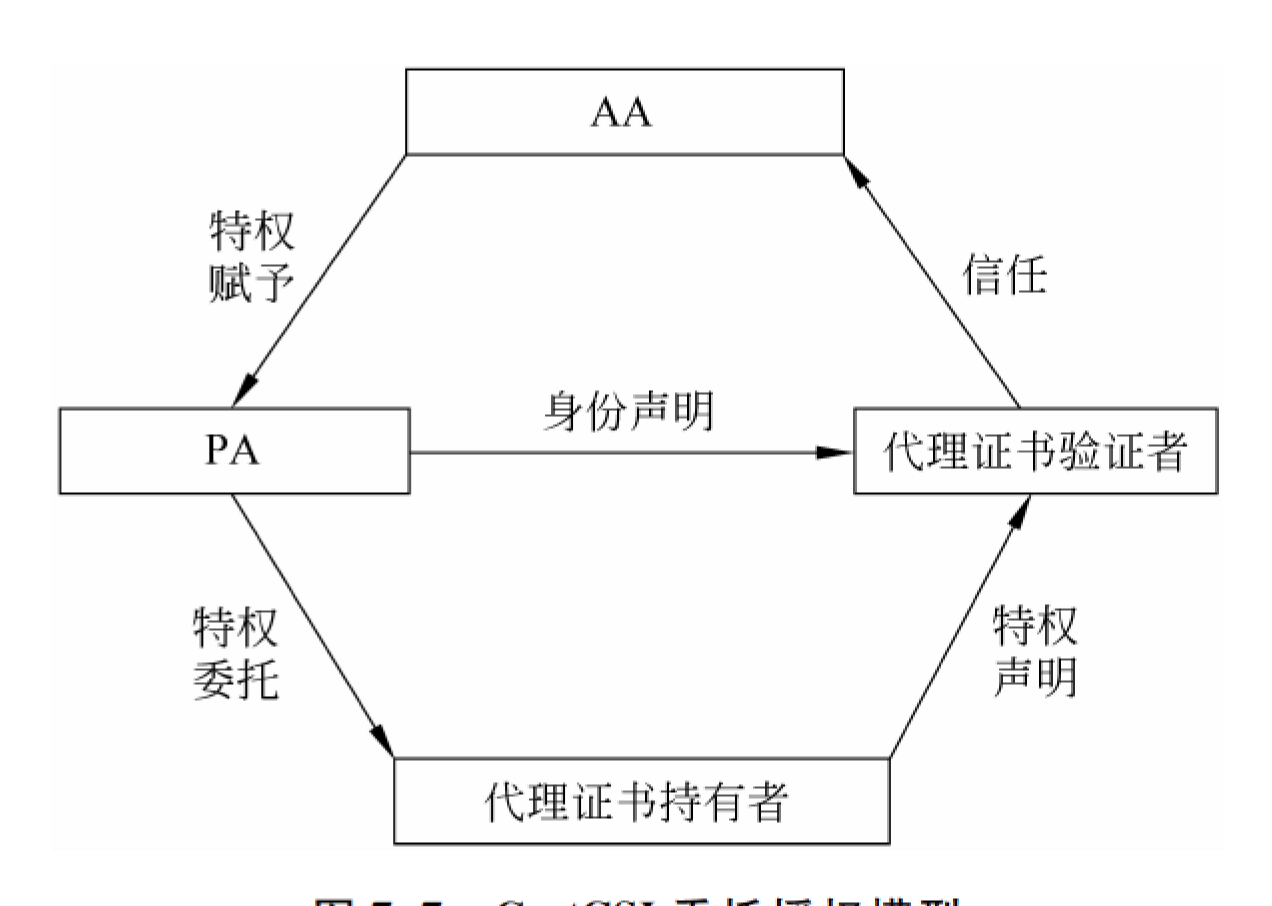


图 7.7 CertGSI 委托授权模型

在 CertGSI 中,资源访问控制机制的具体实现主要由资源代理(RP)及策略引擎(PE)完成。从访问控制策略的角度来看,本地资源一般可以分成下面两种类型:一类是基于远程主体的全局角色(由其出示的证书所反映的身份和属性信息决定)直接进行访问控制;另一类是基于远程主体映射的本地角色进行访问控制,即这类资源的访问控制决策需要将远程主体的全局角色映射为一个本地角色(如本地系统的账号)后才能作出。在 PE 的实现中,对于前一类资源的访问控制采用了"使用条件"列表的技术实现,所谓"使用条件"就是允许远程主体访问资源所必须具备的条件,这种条件是依据远程用户的属性信息制定的。对于后一类资源的访问控制,PE 使用了全局/本地映射表的技术,即根据远程主体出示的属

性证书信息，将它映射成相应的一个本地账号，然后用这个本地账号进行资源访问。可见，通过上述访问控制机制，CertGSI 不但可以兼容不同的本地安全方案，而且其基于策略的授权机制应具有良好的可扩展性，能方便地扩展到具有大量用户和大量资源的开放式环境中。CertGSI 基本的访问控制过程简单描述如下：

(1) 访问请求者(用户、用户代理或者进程)向 RP 发送一个签名的访问请求。

(2) RP 与访问请求者进行双向身份认证。若认证通过则转至下一步，否则本次请求失败。

(3) RP 调用 PE，由 PE 根据访问资源类型及访问请求者的角色(属性证书中的属性信息)进行访问控制决策。

① 如果是第一种资源类型，PE 检查该资源的使用条件列表和访问请求者的属性信息，若符合使用条件则允许访问，否则拒绝访问。

② 如果是第二种资源类型，PE 检查该资源的全局/本地映射表和访问请求者的属性信息，若符合映射条件则将访问请求者映射成一个本地账号进行访问，否则拒绝访问。

7.4 一种基于属性证书的委托授权模型——ACDAM

委托技术是实现网格认证和授权机制的基础。实际上，权力委托是分布式环境下实现灵活授权机制的普遍需求，因此，相关的研究和论文在 20 世纪 90 年代初期开始出现。随着网格技术的出现和发展，文献[3]提出了一种针对网格环境的安全委托机制，它在 PKI 基础上使用代理证书提出了一种基于身份的委托授权机制。这种基于代理证书的委托授权机制可以较好地应用于传统基于身份的授权环境，但不能很好地用于基于属性的授权环境。然而，随着基于角色的访问控制技术以及面向授权管理的 PMI(Privilege Management Infrastructure)技术的应用和发展，目前许多安全系统采用了基于属性的授权机制。与传统基于身份的授权机制相比，基于属性(角色等)的授权机制能更加灵活有效地应用于大规模分布式网络环境中，尤其适合用在具有动态特性的网格环境中。下面深入探讨一种面向网格的基于属性证书的安全委托授权模型，为方便起见，以下简称该模型为 ACDAM。

7.4.1 若干术语与定义

所谓委托(delegation)，指的是权力的委托(delegation of rights)，即一个实体 A 将自己拥有的一部分权力或全部权力授予另一个实体 B，以便 B 通过继承 A 的合法权力后能够访问相应的资源或进行其他的安全操作。A 与 B 之间的这种委托关系可以符号表示为“B for A”。实体 A 将自己拥有的权力委托给实体 B 后，实体 B 根据需要可以将该权力或该权力中的一部分进一步委托给实体 C，这样 A、B、C 之间就形成委托链关系(B for A，C for B)。在一个委托链中，参与委托的各方主体包括以下几种。

发起者(initiator)：是整个委托链的最初授权者和发起者。

委托者(grantor)：也叫委托主体，它将自己的权力授予另一个主体。

受托者(grantee)：也叫受托主体，它接受委托方授予的权力。

验证者(verifier)：也叫服务者，它向委托权力的拥有者提供服务，实施授权决策。

中间者(intermediary)：它是整个委托链中位于发起者和验证者之间的主体。

7.4.2 网格环境下的委托问题

网格技术为人们进行多种形式的资源共享和协同工作提供了条件。大多数网格需要用于一些临时的协作活动中,在这样的网格环境中,其资源和用户组成是动态变化的,因而难以采用传统的授权机制,让管理员事先一一规定好每一个用户对每一个资源所拥有的权限。因此,在这种网格环境中,必须采用一种更灵活的授权机制,即支持权力委托的分布式授权机制。通过这种机制,网格中的主体之间可以互相委托授权,以满足具有大规模、分布和动态等特性的网格环境的授权要求。

为了说明网格环境下安全委托的一些概念,下面讨论图 7.8 给出的一个网格计算例子。

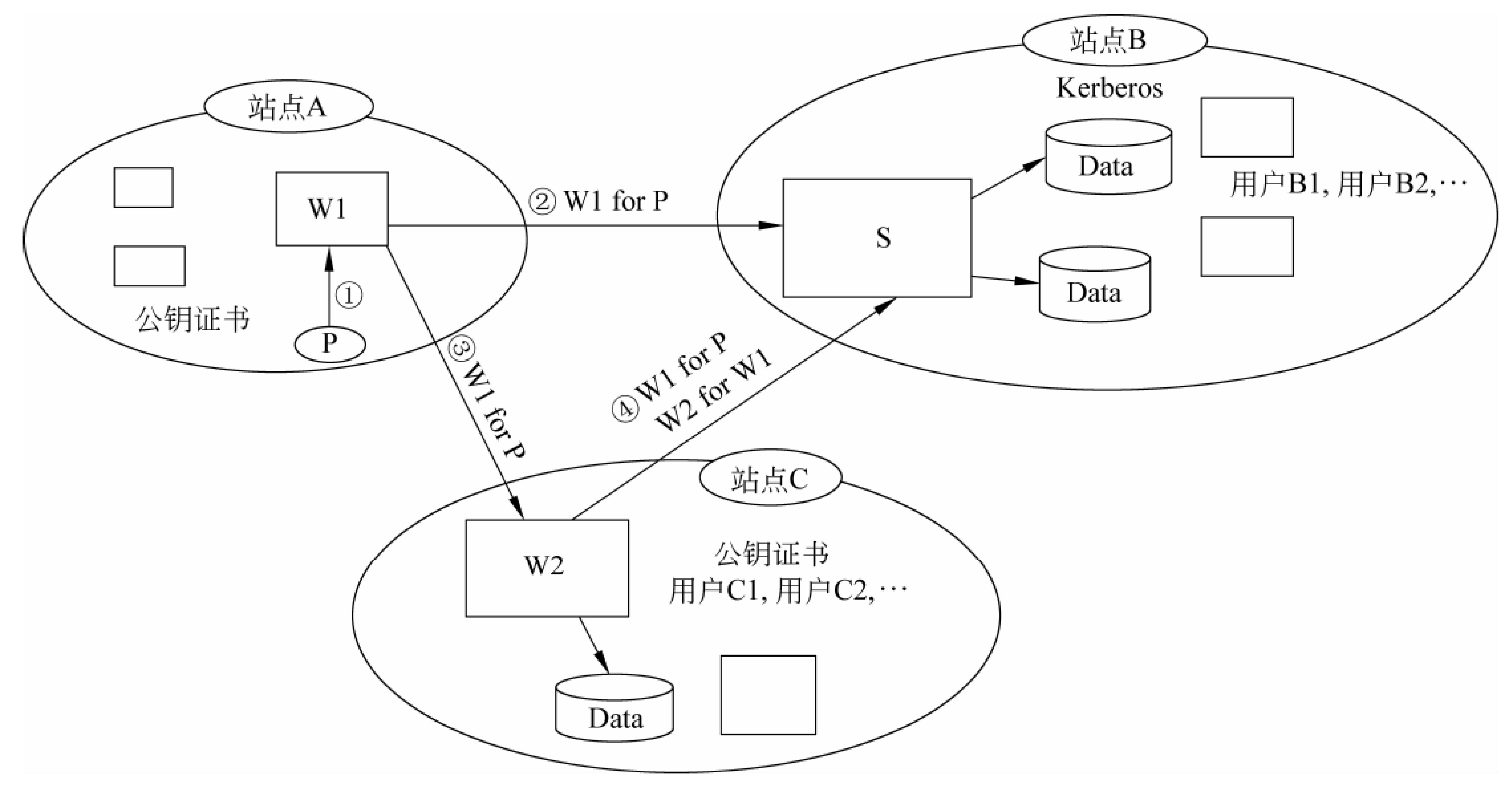


图 7.8 一个网格计算示例

1. 单步委托

图 7.8 中步骤①和②构成了一个单步委托的过程。为了使本地工作站 W1 中的用户代理程序能够代表物理学家 P 启动远程服务器 S 上的物理分析程序,P 需要将自己拥有的运行物理分析程序的权力委托给 W1。这种委托需要一种凭证,以便 W1 向 S 表明这种委托关系(W1 for P),并证明通过委托得到的权力的合法性。

2. 多步委托

图 7.8 中的步骤①、③和④构成了一个长度为 2 的多步委托的过程。为了 W1 中的用户代理程序能够代表 P 启动 W2 上的物理模拟程序,P 需要将自己拥有的运行物理模拟程序的权力委托给 W1,实行一个单步委托(W1 for P)。接下来,远程工作站 W2 还需要代表物理学家 P 访问存放在 S 上的一些用于模拟的参数值,因此它也需要 P 委托相应的权力。然而,假定这个时候 W2 不能与 P 直接交互,只能与 W1 直接交互,所以它只有通过 W1 才能得到 P 的权力。于是,首先 P 将访问参数值的权力委托给 W1,向 W1 颁发相应的委托凭证(W1 for P);然后 W1 再将这个权力进一步委托给 W2,并向 W2 颁发相应的委托凭证(W2 for W1)。当 W2 向 S 发出请求时,通过出示委托凭证链(W1 for P,W2 for W1)证明

自己的委托身份和相应的权力。

为了实现上述单步委托和多步委托，需要提供一套安全可靠的委托机制。利用这种委托机制，权力的受托者能方便地向验证者证实委托的真实性和有效性，避免委托的假冒、伪造和修改；委托者可以控制和约束委托，包括指定可委托的对象、可继承的权力、委托权力的使用范围、委托的有效期以及委托链的长度。另外，委托者根据需要还可以撤销委托。

7.4.3 ACDAM 框架结构

随着面向身份认证的 PKI 技术的普及和发展，目前面向授权管理的 PMI 技术也逐渐在各种安全系统中得到应用。在 PMI 中授权的依据主要是主体的属性，属性是一个主体的特殊性质，用于指明所有者的成员资格、角色、安全许可或者其他一些权力信息。PMI 采用属性证书管理主体的属性，属性证书(AC)用于标明主体具有的属性，它是一种由属性权威(Attribute Authority，AA)签发的将主体与其享有的权力属性绑定在一起的数据结构。与 PKI 中的标识证书(IC)相比，属性证书不包含有实体的公钥，它的生命周期要短，可以灵活地根据主体某种属性的变化而改变。

虽然一个主体的属性信息也可以放在它的标识证书的扩展项中，但是这种方法存在许多局限性。而通过使用属性证书，可以提高访问控制的灵活性与安全性。在具有动态特性的网格环境中，使用属性证书进行授权管理带来的授权灵活性、可扩展性和安全性更加明显。

由于属性证书记录了一个主体的权力信息，因此可以通过属性证书实现权力的传递和委托。ACDAM 就是一种使用属性证书进行权力传递和委托的授权模型，图 7.9 给出了 ACDAM 跨域委托授权的框架。

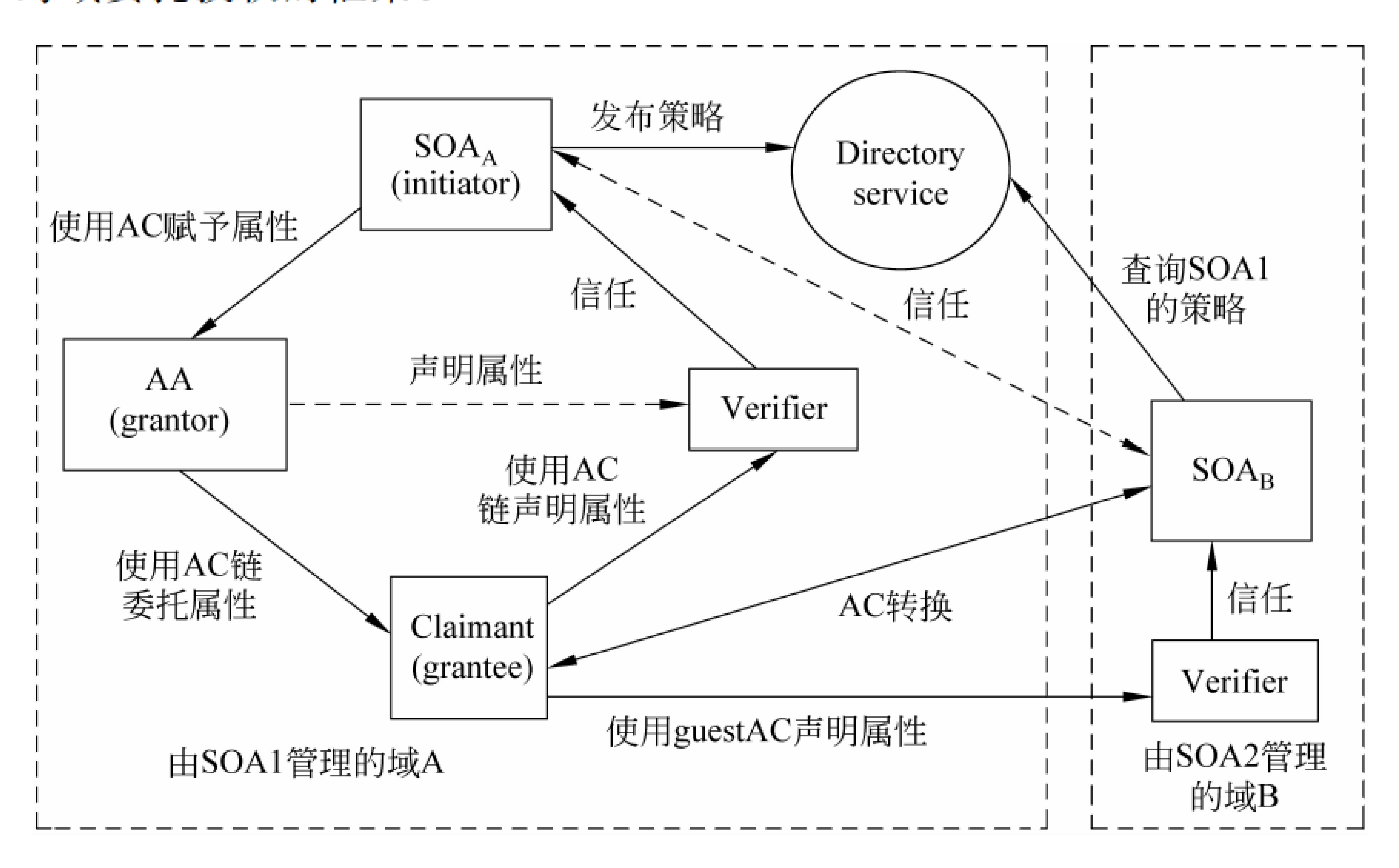


图 7.9 ACDAM 跨域委托框架

从图 7.9 中可以看出，在一个 ACDAM 安全域中包含了 5 种组件：属性管理中心(Source of Authority，SOA)、属性委托者(AA)、属性声明者(Claimant，也是属性受托者)、属性验证者(Verifier)和目录服务(Directory Service)。图中 SOA_A 和 SOA_B 分别是各自所在域的属性管理中心(SOA)，它们通过颁发属性证书将相应的属性赋予最初的属性持有

者。在委托情况下，SOA允许最初的属性持有者充当AA，即通过颁发一个新的AC（作为委托凭证，又可以称为委托证书）将相关属性委托给其他主体。同样，在多步委托的情况下，后续受托主体可以进一步充当AA，将相关属性委托给再下一个受托主体，这样的多步委托便形成一个委托链。在委托链中，前面的AA可以限制后面AA的权力，所有AA委托给下一个主体的权力属性不能大于自己所持有的权力属性。SOA可以约束整个委托链，比如它可以限制委托链的长度，指定后续可受托的主体和属性的有效期等。

属性验证者信任自己所在域的SOA，将SOA视为相应属性的权威。如果属性声明者的属性证书不是由SOA直接颁发的，那么验证者就需要找到从该属性证书追溯到一个由SOA直接颁发的属性证书的委托链，并负责验证委托链中的每一个AA都真正拥有自己声称的属性，并有权委托给其他的主体。

当进行跨域访问时，属性验证者可能无法识别和验证属性声明者的证书，因为两个域之间的安全策略可能不一样。这种跨域访问引起的问题通过采用Directory Service（目录服务）、AC translation（属性证书转换）和guestAC（客人属性证书）解决。当进行跨域委托授权时，验证者所在域的SOA通过目录服务查询获取属性声明者所在域的SOA的安全策略，然后将声明者出示的属性证书链转换成可被验证者识别的单个guestAC。

为了实现受限委托机制，ACDAM使用了一种具有特殊结构的AC，这种AC的内容不同于X.509 PMI中定义的标准AC，它主要包含以下6个字段的内容。

（1）委托主体（Issuer）：包含委托主体标识证书的序列号及签发者。

（2）受托主体（Holder）：包含受托主体标识证书的序列号及签发者。

（3）属性（Attribute）：用于指明证书持有者具有的成员资格、角色、安全许可或者其他一些权力信息。

（4）有效期限（ValidityPeriod）：委托的有效期限。

（5）委托长度（DelegateLength）：受托主体可进一步委托的长度。

（6）证书序列号（SerialNumber）：用于标识该委托主体颁发的每一个委托证书。

7.4.4 ACDAM委托协议

为了描述方便，下面采用一个包含六元组的签名消息表示一个用于委托凭证的属性证书，即

$$\mathrm{DT}_{XY}=<X,Y,P_{XY},T_{XY},L_{XY},N_{XY}>_X$$

其中，X代表委托主体（Issuer），Y代表受托主体（Holder），P_{XY}代表属性（Attribute），T_{XY}代表证书有效期限（ValidityPeriod），L_{XY}代表可进一步委托的长度（MaxPathLength），N_{XY}代表证书序列号（SerialNumber），$<>_X$表示采用委托主体X的私钥对$<>$中的消息进行签名。

1. 域内委托授权过程

图7.10给出了一个长度为2的多步委托过程，委托路径为SOA→A→B→C→S。首先，SOA将相关属性赋予主体A，然后A将属性委托给主体B，B再把相关属性进一步委托给主体C，最后C持委托得到的属性向服务器S发出服务请求。详细描述如下。

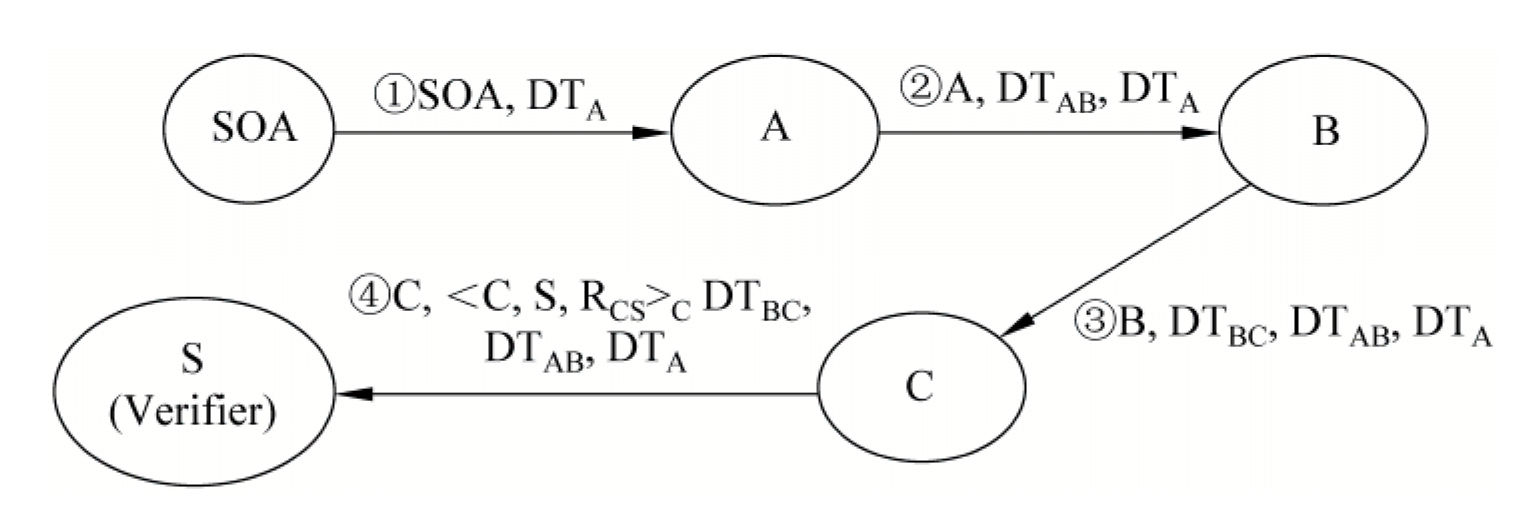


图 7.10　一个域内多步委托授权过程

步骤①：(SOA→A：SOA，DT_A)

其中，$DT_A = <SOA, A, P_A, T_A, L_A, N_A>_{SOA}$

这一步是 SOA 通过向 A 颁发属性证书 DT_A，将属性 P_A 赋予 A。通过 DT_A 的获得，A 拥有属性 P_A。证书是经过 SOA 的私钥 SK_{SOA} 签名的，因此，验证者使用 SOA 的公钥 PK_{SOA} 可以校验 DT_A 内容的真实性和有效性。SOA 通过在 DT_A 中指定证书的有效期限 T_A 和后续委托的最大长度 L_A，可以实现对后续委托链的限制。

步骤②：(A→B：A，DT_{AB}，DT_A)

其中，$DT_{AB} = <A, B, P_{AB}, T_{AB}, L_{AB}, N_{AB}>_A$

$DT_A = <SOA, A, P_A, T_A, L_A, N_A>_{SOA}$

这一步是在上一步的基础上，A 通过向 B 颁发委托证书 DT_{AB}，将属性 P_{AB} 委托给 B。通过 DT_{AB} 的获得，B 拥有属性 P_{AB}。证书 DT_{AB} 是经过 A 的私钥 SK_A 签名的，因此，验证者使用 A 的公钥 PK_A 可以校验内容的真实性。DT_{AB} 中明确指明了委托主体和受托主体，因此可以确保相关属性是从指定主体 A 委托给另一个指定的主体 B，在没有 B 的进一步授权下，任何其他主体持有 DT_{AB} 都是无效的。另外，委托证书 DT_{AB} 中的序列号 N_A 可以用来标识该证书，这便于证书的撤销和维护。

A 向 B 发送的消息中包含两个证书，即 DT_{AB} 和 DT_A。其中，DT_A 是上一步中 SOA 向 A 颁发的属性证书，用以证明 A 确实拥有将属性 P_{AB} 委托给 B 的权力。注意，根据前面的委托原则，为了减少证书被破解后带来的风险，A 向 B 签发的委托证书 DT_{AB} 内容须满足下面 3 个约束条件：

(1) 委托权力约束：$P_{AB} \leqslant P_A$

(2) 委托长度约束：$0 \leqslant L_{AB} < L_A$

(3) 有效期限约束：$T \leqslant T_{AB} \leqslant T_A$（$T$ 指当前时间）

步骤③：(B→C：B，DT_{BC}，DT_{AB}，DT_A)

其中，$DT_{BC} = <B, C, P_{BC}, T_{BC}, L_{BC}, N_{BC}>_B$

$DT_{AB} = <A, B, P_{AB}, T_{AB}, L_{AB}, N_{AB}>_A$

$DT_A = <SOA, A, P_A, T_A, L_A, N_A>_{SOA}$

这一步是在前面两步的基础上，B 通过向 C 签发委托证书 DT_{AB}，进一步将相关属性(P_{BC})委托给 C。DT_{BC} 证书是经过 B 的私钥 SK_B 签名的，因此，验证者使用 B 的公钥 PK_B 可以校验内容的真实性。A 向 B 发送的消息中包含 3 个证书，即 DT_{BC}、DT_{AB} 和 DT_A。其中，DT_{AB} 和 DT_A 是前面两步中签发的证书，用以证明该委托的完整来源和合法性。

根据受限委托的原则，B 向 C 签发的委托证书 DT_{BC} 的内容必须满足下面 3 个约束

条件：

(1) 委托权力约束：$P_{BC} \leqslant P_{AB}$

(2) 委托长度约束：$0 \leqslant L_{BC} < L_{AB}$

(3) 有效期限约束：$T \leqslant T_{BC} \leqslant T_{AB}$

步骤④：(C→S：C，$<C, S, R_{CS}>$，SK_C，DT_{BC}，DT_{AB}，DT_A)

其中，$DT_{BC} = <B, C, P_{BC}, T_{BC}, L_{BC}, N_{BC}>_B$

$DT_{AB} = <A, B, P_{AB}, T_{AB}, L_{AB}, N_{AB}>_A$

$DT_A = <SOA, A, P_A, T_A, L_A, N_A>_{SOA}$

这一步是主体 C 持委托得到的权力属性 P_{BC} 向服务器 S 发出签名的资源访问请求 R_{CS}。C 向 S 发送的消息中包含与委托相关的完整证书链(3 个证书，即 DT_{BC}、DT_{AB} 和 DT_A)，用以证明自己拥有的委托身份和权力属性。S 在作出授权决策之前，必须一一验证 DT_{BC}、DT_{AB} 和 DT_A 的真实性和有效性，以证实请求者 C 的委托身份，并验证 A 确实拥有权力将属性 P_{AB} 委托给 B，而 B 拥有权力将属性 P_{BC} 委托给 C。同时，S 还需要检查下面 3 个委托约束条件是否满足：

(1) 委托权力约束：$P_{BC} \leqslant P_{AB} \leqslant P_A$

(2) 委托长度约束：$0 \leqslant L_{BC} < L_{AB} < L_A$

(3) 有效期限约束：$T \leqslant T_{BC} \leqslant T_{AB} \leqslant T_A$

通过上述验证后，S 就可以依据权力属性 P_{BC} 和相关资源的授权策略作出授权决策，或准许访问，或拒绝访问。

2. 跨域委托授权过程

图 7.11 给出了一个跨域多步委托过程，委托路径为 SOA1→A→B→C→SOA2→C→S，它可以分成 6 个步骤：

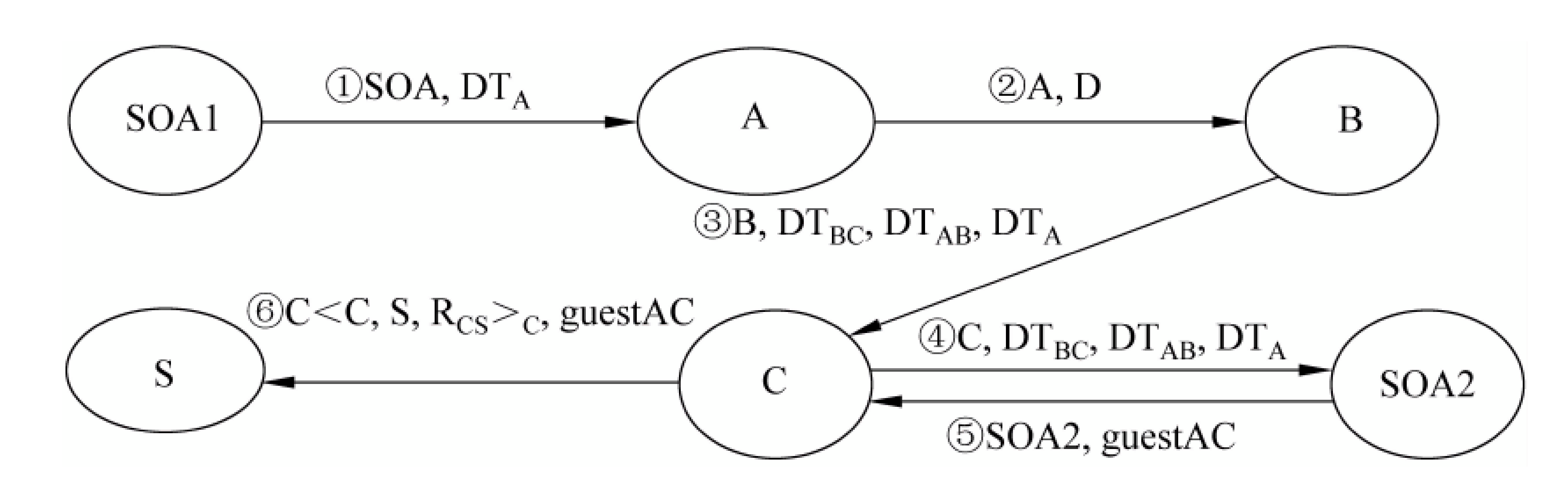


图 7.11 一个跨域多步委托授权过程

(1) 属性声明者所在域的属性管理中心 SOA1 将相关属性赋予主体 A(SOA1→A)；

(2) A 将相关的属性委托给主体 B(A→B)；

(3) B 再把相关属性进一步委托给主体 C(B→C)；

(4) C 向验证者所在域的属性管理中心 SOA2 发出请求，将属性证书链转换为可被服务器 S 识别的证书(C→SOA2)；

(5) SOA2 授予 C 一个客人属性证书 guestAC(SOA2→C)；

(6) C 持 guestAC 向服务器 S 发出服务请求(C→S)。

因为步骤①、步骤②和步骤③与前述域内委托过程完全一样，所以下面只简要补充描述步骤④、步骤⑤和步骤⑥。

步骤⑤：(C→SOA2：C，DT_{BC}，DT_{AB}，DT_{A})

其中，$DT_{BC}=<B,C,P_{BC},T_{BC},L_{BC},N_{BC}>_{B}$

$DT_{AB}=<A,B,P_{AB},T_{AB},L_{AB},N_{AB}>_{A}$

$DT_{A}=<SOA1,A,P_{A},T_{A},L_{A},N_{A}>_{SOA}$

约束条件：

(1) $P_{BC}\leqslant P_{AB}\leqslant P_{A}$

(2) $0\leqslant L_{BC}<L_{AB}<L_{A}$

(3) $T_{BC}\leqslant T_{AB}\leqslant T_{A}$

步骤⑥：(SOA2→C：SOA2，guestAC)

其中，$guestAC = <SOA2,C,P_{C2},T_{C2},L_{C2},N_{C2}>_{SOA2}$

步骤⑦：(C→S：C，$<C,S,R_{CS}>_{C}$，guestAC)

其中，$guestAC = <SOA2,C,P_{C2},T_{C2},L_{C2},N_{C2}>_{SOA2}$

7.5 一种支持信任管理的委托授权模型——TrustDAM

现有的网格安全技术大多数是传统PKI技术的改进和扩充，实质上只是暂时解决了身份认证等一些初步的安全问题，难以满足不断发展中的网格应用环境的分布异构性、多变性和不确定性的要求。如前所述，“信任管理”和“信任协商”等思想方法的出现，为所有基于开放、分布、动态特性环境的安全问题提供了新的解决思路，同时也为网格安全问题的解决提供了新方法。下面在前一节内容的基础上，专门探讨一种支持信任管理的委托授权模型——TrustDAM。

7.5.1 网格环境下的信任管理问题

第1章已指出“信任”是一个非常复杂的概念，目前国内外对于信任都还没有一个精确的、广泛可接受的定义。实际上，信任的含义是与具体的应用环境相关的。我们认为，在网格应用环境中，凭证信任和行为信任是两种最主要的信任形式，它们对网格的安全保障都是缺一不可的因素。

1. 基于凭证的网格信任管理问题

目前在信息安全领域中，现有的大多数信任模型一般都是通过某种安全凭证（如标识和证书等）建立实体之间的信任关系。例如，在PKI体系中采用身份证书建立身份信任关系，在PMI体系中则采用属性证书建立属性（角色和能力等）信任关系。我们将这一类关系统称为“基于凭证的信任关系”。它们涉及的是一类可精确描述和推理的、度量绝对的、静态的信任关系。

在网格环境中，目前广泛使用的GSI安全机制以及GGF提出的基于CA的信任模型实质上只是解决了身份信任的问题，并没有涉及属性信任关系，不支持灵活的基于策略的授权机制。而且它们在支持身份认证的安全委托和可扩展性等方面也还有许多局限性。

2. 基于行为的网格信任管理问题

基于凭证的信任管理通过各种安全凭证的使用可以基本解决身份信任、角色/能力信任

等问题,但是它并不能解决所有的网格安全问题。例如,一个需要使用远程大型计算机运行其程序的网格用户,尽管采用证书通过了系统的身份和角色验证,但是这并不排除他还有恶意的行为,或者他运行的程序存在风险。这正如在现实社会中一个有身份、有资格的人,并不意味着他的行动就是完全可信赖的一样。因此,在网格环境中还需要研究行为信任的问题。如前所述,所谓基于行为的信任管理,就是通过实体的行为表现来度量实体的信任度,并依此建立实体之间的信任关系。我们认为,在网格这样复杂的应用环境中实施信任管理技术,关键的环节在于信任信息的获取和反馈机制,以及与安全策略实施相结合的机制,而不仅仅在于信任度评估数学模型的建立。

7.5.2 TrustDAM 框架结构

为了在 ACDAM 的基础上建立一个完全支持信任管理的委托授权模型 TrustDAM,首先需要明确"信任"(trust)和"声誉"(reputation)这两个概念。

如前所述,"信任"是一个非常模糊的概念,目前国内外对于信任都还没有一个精确的、广泛可接受的定义。为了方便起见,在 TrustDAM 中采用如下定义[4]:"Trust is the firm belief in the competence of an entity to act as expected such that this firm belief is not a fixed value associated with the entity but rather it is subject to the entity's behavior and applies only within a specific context at a given time."

我们使用信任值(Trust Value,TV)度量信任的程度,信任值是一个动态的值,如果以信任程度的百分比来表示,它的取值范围为 0～1 的实数,1 表示完全信任(fully trustworthy),0 表示完全不信任(fully untrustworthy)。

可以根据一个实体的声誉评估它的信任程度,在 TrustDAM 中也使用文献[4]给出的"声誉"定义:"The reputation of an entity is an expectation of its behavior based on other entities' observations or the collective information about the entity's past behavior within a specific context at a given time."

为了实现更加安全的委托授权机制,TrustDAM 使用了一种带有信任值 TV 的属性证书 AC,这种 AC 主要包含以下 7 个字段的内容。

(1) 委托主体(Issuer):包含委托主体标识证书的序列号及签发者。

(2) 受托主体(Holder):包含受托主体标识证书的序列号及签发者。

(3) 属性(Attribute):用于指明证书持有者具有的成员资格、角色、安全许可或者其他一些权力信息。

(4) 信任值(TrustValue):委托主体赋予该委托的信任程度。

(5)有效期限(ValidityPeriod):委托的有效期限。

(6)委托长度(DelegateLength):受托主体可进一步委托的长度。

(7)证书序列号(SerialNumber):用于标识该委托主体颁发的每一个委托证书。

图 7.12 给出了 TrustDAM 部署在一个网格虚拟组织环境中的框架。与 ACDAM 相比,TrustDAM 由于集成了信任管理功能,所以能达到更加安全的委托授权机制。服务器可以通过检查委托证书链中的信任值 TV,评估该委托链的可靠程度。TV 是通过实体的声誉(reputation)和实体之间的直接信任关系计算得到的,而声誉是通过域属性管理中心 SOA 提供的声誉服务(Reputation Service,RS)查询得到的;实体之间的直接信任关系根据

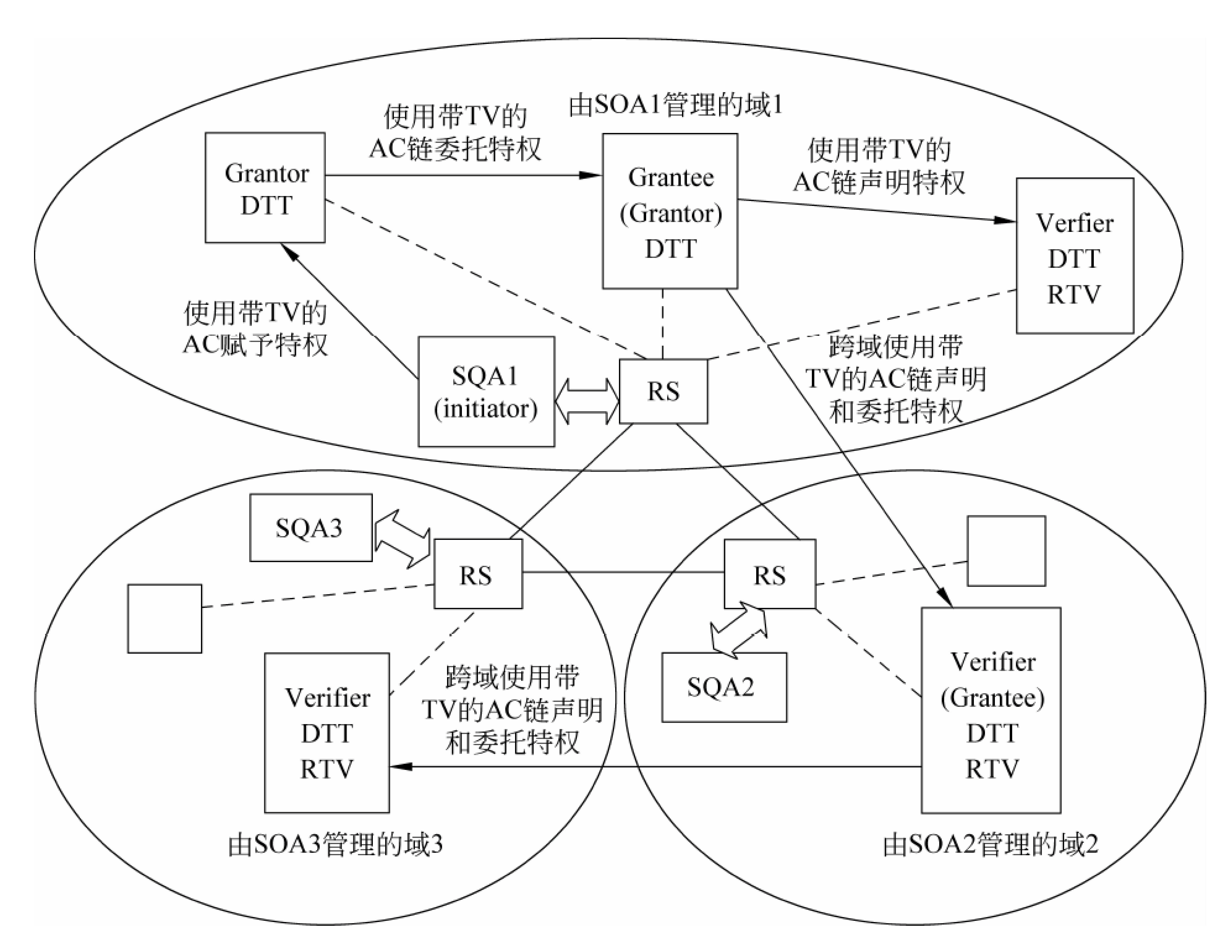


图 7.12　TrustDAM 在一个网格虚拟组织中的框架结构

每一个实体自己维护的直接信任表(Direct Trust Table,DTT)计算。服务器在进行基于信任的授权决定时,可以预先设定一个信任需求值(Required Trust Value,RTV),如果委托证书链的信任值小于该信任需求值 RTV,就拒绝提供服务。同样,服务器也可以设定一个声誉需求值(required reputation value,RRV),如果委托证书链的声誉值小于该声誉需求值(RRV),就拒绝提供服务。

7.5.3　信任和声誉的计算方法

1. 符号与定义

(1) 假设 D_i 和 D_j 代表网格环境中的两个域。

(2) $\Gamma(D_i, D_j, t)$代表在 t 时刻 D_i 和 D_j 关于委托的信任关系。

(3) $\theta(D_i, D_j, t)$代表在 t 时刻 D_i 对 D_j 关于委托的直接信任关系。

(4) $\Omega(D_j, t)$代表在 t 时刻 D_j 关于委托的声誉。

(5) DTT(D_i, D_j)代表 D_i 对 D_j 关于委托的直接信任表。

$\gamma(t - t_{ij})$是一个时间衰减函数,它代表两个实体之间的信任值随着时间衰减的程度,t 是当前时间,t_{ij} 表示 D_i 和 D_j 之间最近一次发生接触的时刻或 DTT 最近被更新的时刻。

2. 声誉的计算和评估

在图 7.12 给出的网格环境中,由每个域中的 SOA 负责提供声誉服务(RS),RS 需要不断更新和维护本域内各个实体的声誉值,以及负责计算、维护和发布其他域的声誉值。

1) 域声誉的计算

域 D_j 的声誉值按如下公式计算:

$$\Omega(D_j,t)=\frac{\sum_{k=1}^{n}\mathrm{DTT}(D_k,D_j)\times R(D_k,D_j)\times\gamma(t-t_{kj})}{\sum_{k=1}^{n}D_k} \tag{7.1}$$

其中，$k\neq j$，$R(D_k,D_j)$是推荐者的信任因数，是为了防止小范围内联合欺骗行为出现而引入的，它与两实体间的相互熟悉程度成反比，两实体间越熟悉，或隶属于同一联盟，推荐信任因数值越低，取值范围为 0～1 之间的实数。

2）实体声誉的计算

一个域内实体 E_j 的声誉值按如下公式计算：

$$\Omega(E_j,t)=\frac{\sum_{k=1}^{n}\mathrm{DTT}(E_k,E_j)\times R(E_k,E_j)\times\gamma(t-t_{kj})}{\sum_{k=1}^{n}E_k} \tag{7.2}$$

其中，$k\neq j$，E_k 和 E_j 代表同一个域内的两个实体，其他符号的含义同式(7-1)。

3）委托链声誉值的评估

（1）同一个域内的委托链

假定委托链的委托路径为 $A\rightarrow E_1\rightarrow E_2\rightarrow\cdots\rightarrow E_n\rightarrow B$，其中 A 是发起者，B 是验证者，其他是在同一个域内的中间者。那么，该委托链声誉值的计算公式如下：

$$\mathrm{RV}(\mathrm{DC},t)=\Omega(A,t)\times\prod_{j=1}^{n}\Omega(E_j,t) \tag{7.3}$$

（2）跨域的委托链

假定一个委托链的委托路径为 $A\rightarrow D_1\rightarrow D_2\rightarrow\cdots\rightarrow D_n\rightarrow B$，其中 A 发起者，B 是验证者，其他是在 n 个不同域内的中间者。那么，该委托链声誉值的计算公式如下：

$$\mathrm{RV}(\mathrm{DC},t)=\Omega(A,t)\times\prod_{j=1}^{n}\Omega(D_j,t) \tag{7.4}$$

3. 信任的计算和评估

1）信任值的计算

一个属性证书 AC 上的信任值（TV）代表了委托主体（设为 E_i）对受托主体（设为 E_j）的信任关系，因此信任值 TV 按如下公式计算：

$$\mathrm{TV}=\Gamma(E_i,E_j,t)=\alpha\times\theta(E_i,E_j,t)+\beta\times\Omega(E_j,t) \tag{7.5}$$

其中，$\alpha,\beta\geqslant 0$，$\alpha+\beta=1$。

$\theta(E_i,E_j,t)$ 代表在 t 时刻 E_i 对 E_j 关于委托的直接信任关系，因此它的计算公式如下：

$$\theta(E_i,E_j,t)=\mathrm{DTT}(E_i,E_j)\times\gamma(t-t_{ij}) \tag{7.6}$$

$\Omega(E_j,t)$ 代表在 t 时刻 E_j 的声誉值。如果 E_i 和 E_j 在同一个域内，$\Omega(E_j,t)$可以按照式(7-2)计算。如果 E_i 和 E_j 不在同一个域内，可以将 E_j 所在域 D_j 的声誉值看作是 E_j 声誉值，那么 $\Omega(E_j,t)$就可以按照式(7-1)计算。

2）委托链信任值的评估

假定一个委托链的委托证书路径为 $\mathrm{AC}_1\rightarrow\mathrm{AC}_2\rightarrow\cdots\rightarrow\mathrm{AC}_n$，各证书的信任值为：$\mathrm{TV}_1$ for AC_1，TV_2 for AC_2，…，TV_n for AC_n。那么，委托链的整体信任值 TV(DC)按以下公式计算：

$$\mathrm{TV}(\mathrm{DC})=\prod_{j=1}^{n}\mathrm{TV}_j \tag{7.7}$$

7.5.4 TrustDAM 委托协议

下面采用一个包含七元组的签名消息表示一个用于 TrustDAM 委托凭证的属性证书，即

$$\mathrm{DT}_{XY}=<X,Y,P_{XY},T_{XY},L_{XY},\mathrm{TV}_{XY},N_{XY}>_X$$

其中，X 代表委托主体(Issuer)，Y 代表受托主体(Holder)，P_{XY}代表属性(Attribute)，T_{XY}代表证书有效期限(ValidityPeriod)，L_{XY}代表可进一步委托的长度(MaxPathLength)，TV_{XY}代表信任值(TrustValue)，N_{XY}代表证书序列号(SerialNumber)，$<>_X$ 表示采用委托主体 X 的私钥对$<>$中的消息进行签名。

1. 委托证书的创建

委托主体 A 与受托主体 B 之间进行委托的过程中，需要创建一个用作委托凭证的带有信任值的属性证书，其步骤如下。

步骤 1：委托主体 A 从自己维护的 DTT 中读取 DTT(A,B)，然后按式(7.6)计算 A 和 B 之间的直接信任关系 $\theta(\mathrm{A},\mathrm{B},t)$。

步骤 2：委托主体 A 查询自己所在域的声誉服务(RS)，获取受托主体 B 的声誉值 $\Omega(E_j,t)$。

步骤 3：委托主体 A 按式(7.5)计算该委托的信任值 $\mathrm{TV}_{\mathrm{AB}}$。

步骤 4：委托主体 A 使用自己的私钥创建一个包含信任值 $\mathrm{TV}_{\mathrm{AB}}$等内容的委托证书 $\mathrm{DT}_{\mathrm{AB}}=<\mathrm{A},\mathrm{B},P_{\mathrm{AB}},T_{\mathrm{AB}},L_{\mathrm{AB}},\mathrm{TV}_{\mathrm{AB}},N_{\mathrm{AB}}>_{\mathrm{A}}$。

2. 委托授权过程

TrustDAM 的链式委托过程与 7.4.4 节中介绍的 ACDAM 链式委托过程基本一样，这里不再赘述。下面只简要地给出图 7.10 所示的一个域内链式委托过程的形式化描述。

步骤 1：SOA→A：SOA,DT_{A}

其中，$\mathrm{DT}_{\mathrm{A}}=<\mathrm{SOA},\mathrm{A},P_{\mathrm{A}},T_{\mathrm{A}},L_{\mathrm{A}},\mathrm{TV}_{\mathrm{A}},N_{\mathrm{A}}>_{\mathrm{SOA}}$

步骤 2：A→B：A,$\mathrm{DT}_{\mathrm{AB}}$,$\mathrm{DT}_{\mathrm{A}}$

其中，$\mathrm{DT}_{\mathrm{AB}}=<\mathrm{A},\mathrm{B},P_{\mathrm{AB}},T_{\mathrm{AB}},L_{\mathrm{AB}},\mathrm{TV}_{\mathrm{AB}},N_{\mathrm{AB}}>_{\mathrm{A}}$

$\mathrm{DT}_{\mathrm{A}}=<\mathrm{SOA},\mathrm{A},P_{\mathrm{A}},T_{\mathrm{A}},L_{\mathrm{A}},\mathrm{TV}_{\mathrm{A}},N_{\mathrm{A}}>_{\mathrm{SOA}}$

约束条件：

(1) $P_{\mathrm{AB}}\leqslant P_{\mathrm{A}}$

(2) $0\leqslant L_{\mathrm{AB}}<L_{\mathrm{A}}$

(3) $T_{\mathrm{AB}}\leqslant T_{\mathrm{A}}$

步骤 3：B→C：B,$\mathrm{DT}_{\mathrm{BC}}$,$\mathrm{DT}_{\mathrm{AB}}$,$\mathrm{DT}_{\mathrm{A}}$

其中，$\mathrm{DT}_{\mathrm{BC}}=<\mathrm{B},\mathrm{C},P_{\mathrm{BC}},T_{\mathrm{BC}},L_{\mathrm{BC}},\mathrm{TV}_{\mathrm{BC}},N_{\mathrm{BC}}>_{\mathrm{B}}$

$\mathrm{DT}_{\mathrm{AB}}=<\mathrm{A},\mathrm{B},P_{\mathrm{AB}},T_{\mathrm{AB}},L_{\mathrm{AB}},\mathrm{TV}_{\mathrm{AB}},N_{\mathrm{AB}}>_{\mathrm{A}}$

$\mathrm{DT}_{\mathrm{A}}=<\mathrm{SOA},\mathrm{A},P_{\mathrm{A}},T_{\mathrm{A}},L_{\mathrm{A}},\mathrm{TV}_{\mathrm{A}},N_{\mathrm{A}}>_{\mathrm{SOA}}$

约束条件：

(1) $P_{BC} \leqslant P_{AB}$

(2) $0 \leqslant L_{BC} < L_{AB}$

(3) $T_{BC} \leqslant T_{AB}$

步骤 4：$C \rightarrow S$：$C, <C,S,R_{CS}>C, DT_{BC}, DT_{AB}, DT_A$

其中，$DT_{BC} = <B,C,P_{BC},T_{BC},L_{BC},TV_{BC},N_{BC}>_B$

$DT_{AB} = <A,B,P_{AB},T_{AB},L_{AB},TV_{AB},N_{AB}>_A$

$DT_A = <SOA,A,P_A,T_A,L_A,TV_A,N_A>_{SOA}$

R_{CS}代表 C 向 S 发出的请求。

约束条件：

(1) $P_{BC} \leqslant P_{AB} \leqslant P_A$

(2) $0 \leqslant L_{BC} < L_{AB} < L_A$

(3) $T_{BC} \leqslant T_{AB} \leqslant T_A$

7.6 本章小结

本章在分析网格安全需求的基础上，探讨了可支持安全委托的网格认证和授权问题，提出了一种新的基于多种证书的网格安全系统 CertGSI。该系统通过灵活使用标识证书、属性证书和代理证书等多种不同用途的数字证书，不但可以满足网格环境下的各种安全需求，而且它提供的认证和授权机制具有较强的灵活性和安全性。本章也深入分析了网格环境下的安全委托问题，提出了一种基于属性证书的安全委托授权模型——ACDAM，该模型通过一种用作委托凭证的属性证书实现权力属性的委托和传递，使用委托证书链实现多步委托。作为信任管理思想方法的一种应用，本章提出了一种支持信任管理的委托授权模型——TrustDAM。该模型建立在 ACDAM 基础上，两者的不同之处在于，TrustDAM 的委托证书中附加携带了一个信任值(Trust Value，TV)，并且增添了关于域和实体声誉(reputation)评估和计算的相关组件和服务。因此，在 TrustDAM 中，服务器可以通过验证委托证书链的信任值以及评估委托证书链的声誉值以加强委托授权的安全性和可靠性。

参考文献

[1] I. Foster, C. Kesselman, The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers, 1999.

[2] S. Farrell, R. Housley. An Internet Attribute Certificate Profile for Authorization. RFC3281, April 2002.

[3] S. Tuecke, D. Engert, I. Foster. Internet X.509 Public Key Infrastructure Proxy Certificate Profile. Internet Draft, August 2001.

[4] Farag Azzedin, Muthucumaru Maheswaran. Evolving and Managing Trust in Grid Computing Systems. In: Canadian Conference on Electrical and Computer Engineering 2002, May 2002: 1424-1429.

[5] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke. A Security Architecture for Computational Grids. In: Proc. 5th ACM Conference on Computer and Communications Security Conference, 1998: 83-92.

[6] 徐锋,吕建. Web 安全中的信任管理研究与进展. 软件学报,2002,11.
[7] 唐文,陈钟. 基于模糊集合理论的主观信任管理模型研究. 软件学报,2003,8.
[8] 蒋文保,戴一奇,杨大鉴. 一种新的网格环境下的安全系统. 清华大学学报(自然科学版),2004,4.
[9] Wenbao Jiang, Hua Song, Yiqi Dai. Realizing Restricted Delegation Using Attribute Certificates in Grid Environments. In: 5th International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD2004), June 2004.
[10] Wenbao Jiang, Chen Li, Shuang Hao, Yiqi Dai. Using Trust for Restricted Delegation in Grid Environments. Lecture Notes in Computer Science (Springer), 2005, Volume 3439: 293-301.
[11] Fan Yong, MA Mei, Jiang Wenbao. FarMon: An Extensible, Efficient Cluster Monitoring System. In: International Conference on Computing in High Energy and Nuclear Physics 2001 (CHEP'01), September, 2001.
[12] 蒋文保. 网格环境下的安全技术研究. 中国科学院研究生院博士学位论文,2002.
[13] 蒋文保. 支持委托的网格认证与授权研究. 清华大学博士后研究报告,2005.

第 8 章　信任管理与网络诚信建设

随着计算机和网络在人类生活的各个领域中的广泛应用,网络在给人类带来了种种便利的同时,也带来了种种挑战和危机。目前,计算机病毒、木马、蠕虫、网络钓鱼以及利用网络窃取国家和商业秘密、传播反动和黄色淫秽信息、侵犯个人隐私等计算机犯罪案件不断增多,已经严重威胁到个人利益和国家安全,给互联网带来的损失也日益增加。因此,网络诚信和安全问题日益凸显。如何规范网络行为,如何打击网络的违法犯罪活动,如何鼓励网络诚信行为,成为当前的热门话题。为了管理和规范网络主体的行为,加强网络诚信建设,必须建立一套有效的网络主体信任评价标准、评价方法和评价体系。

8.1　网络诚信概述

当前,互联网已经成为我国信息社会的重要基础设施,成为人们工作和生活的重要工具。但是,在互联网行业蓬勃发展的过程中,社会上的各种不诚信现象也在互联网上发生,既损害了互联网行业的社会形象,又阻碍了互联网行业的发展。

所谓网络诚信,是指网络行为主体在进行的所有网络行为中,诚实守信,言行一致,不欺骗别人,不发布虚假信息,不侵犯其他网络主体的权利,所有行为力求真实可靠,不利用网络作为工具从事一切不诚信的行为[1]。网络诚信是网络环境下建立在人与人交往基础上的网络虚拟社会的道德问题,是现实社会诚信问题在网络虚拟社会的延伸,是网络虚拟社会存在的价值基础和发展的动力之源。

网络诚信建设是一项系统工程,需要社会各方面的参与和支持,从管理、法律、道德和技术等多个环节建设,才能建立一套符合社会发展需要的网络诚信体系。在网络诚信体系建设中,为了管理和规范网络主体的行为,必须建立一套有效的网络主体信任评价标准、评价方法和评价体系。我们认为,网络主体主要包括网络用户、网站业务经营者、网络服务商和软件等。本章主要讨论软件信任评价体系、网站信任评价体系和网络个人用户信任评价体系。

8.2　软件信任评价体系

本节提出一种基于 AHP 的软件信任评价模型(AHP-based Trust Evaluation Model for Software,ATEMS),来评价和保证软件的可信性。软件信任评价模型包括构造判断矩阵、计算权重值、一致性检验以及信任度贡献值确定和信任度的计算。本节为了检验软件信任评价模型,应用两种软件(IM 软件 QQ 和下载软件迅雷)对模型进行实例分析,以检测模型的可用性。经检验,评价模型的结论与调查的结论相同,从而验证了本节模型的可用性。

8.2.1 软件信任评价

由于目前网络木马和流氓软件等恶意软件的蔓延严重影响了人们各个方面的活动，甚至造成巨大的经济损失。因此，建立一种评价软件可信性的评价体系对社会活动的进行和构建良好的网络环境都具有重要的作用。本节提出建立一套以政府部门作为第三方来对软件的信任进行评价并制定标准、运用 AHP 方法建立评价模型来对软件的具体信任度进行评价的软件信任评价体系。

1. 软件信任评估机构

由谁来评估软件是首先需要解决的问题。软件的评估不能像一些简单的商品那样可以让使用客户来评价，因为软件本身就是由计算机语言编写成的，软件的最终返回结果以及可视化部分都是软件的设计者想让我们看到的，并且经过软件本身或其他过程加工过的结果。如果要评价软件的好坏，需要非常多的其他辅助软件的帮助，才能知道软件是否在使用的过程中与互联网连接并收发数据，是否修改注册表，是否开启了未知的端口，是否捆绑了木马，是否带有病毒，等等。就算通过许多的辅助软件也不一定能够确定软件的可信性，比如其携带的病毒或其本身会设置一定的条件以触发某一事件，在该事件没有发生之前是无法知道这一功能的。

所以，要彻底地评估软件的好坏，就要得到软件的源代码，这样才能知道软件各个部分、各个模块是如何运行的，它运行的原理是什么，是以什么为基础运行的，等等。但是软件的源代码以及编程思路是作为专利申请过的，是对外保密的。用户对使用的软件处在信息不对称的地位，故其评价可以作为参考，但是权重应该很小；如果由第三方来评价软件的可信性是比较理想的，并且很多的律师以及社会舆论都比较赞同这一做法。但是还有一个问题存在，即第三方的可信性又该如何评价。第三方的可信性直接决定了软件的可信性，软件的可信性是决定着网络用户对互联网和自己的计算机的正常使用。政府部门作为第三方有可信性的保证，也就保证了对软件可信性评估的可信性。公司及其个人想要在网络上推行自己的软件，就应该由政府部门作为第三方对其进行评估，并公开评估之后的结果。这种结果可以以新闻的形式张贴在该部门的网站上。同时，该政府部门为软件颁发证书(可以使用一段数字，也可以使电子证书等形式，用户在该部门网站上可以查询证书)，具有证书的软件才能得到国家有关部门认可，这类似于质量认可中的产品质量认证的 ISO 认证和 IEC 认证。同时，我们还需要设计出适合我国国情的软件可信性管理体系，如质量管理体系中的 ISO 9000 质量体系认证。除了上述体系和相关的评估模型外，还要辅以法律的配合。出台相关的法律法规和建立软件信用数据库，规定通过认证的软件是可信软件，并规定认证软件和非认证软件的赔偿问题。

2. 软件信任评价方法

由于应用软件用途广泛，使用群体庞大，所涉及的因素众多，使得对软件的评价很难确定统一的标准。软件的可信性随软件用途的变化而变化，并且软件的设计是面向使用者的。因此，基于使用者评价数据构造软件的可信性模型，既解决了统一标准过于粗略造成的评价数据不准确的问题，又使评价结果更具人性化。层次分析法(Analytic Hierarchy Process, AHP)通过采集使用者评价数据构造两两比较判断矩阵，通过处理判断矩阵得出各因素的

权重值。相同的模型,不同的使用者数据和不同的软件有着不同的权重值。同时,层次分析法把研究对象作为一个系统,把定性和定量方法相结合,并且实施步骤和计算均较为简便,可称得上是从使用者出发的适用于广大用户的方法。

基于此,本节提出了一种新的软件信任评价模型,即一种基于 AHP 的软件信任评价模型(AHP-based Trust Evaluation Model for Software,ATEMS)。第一,它按层次分析法要求建立 3 层层级模型:目标层、准则层和子准则层;第二,根据各因素间的相互比较数据(由调查问卷得到)构造比较矩阵;第三,计算权重值(即为归一化的最大特征值对应的特征向量)并对矩阵进行一致性检验;第四,确定子准则层的信任度贡献值;第五,计算结果。

3. 软件信任度的影响因素

影响软件信任度的因素众多,本节将影响因素划分为 5 个方面,包括设计中缺陷、源代码、操作系统、功能漏洞以及涉及的相关内容。

1) 设计中的缺陷

对于软件来说,其开发设计过程是非常重要的,在开发设计中决定了软件的编写语言、功能用途、使用的操作系统、图形界面以及一些对于计算机系统内部命令和文件的调用等软件应该具有的属性。

2) 源代码是否开放

源代码是否开放也是决定软件信任度的重要因素。有人可能认为开源的软件是安全的,因为所有人都可以看到软件的源代码,从而知道软件是如何运行的、调用了操作系统的什么文件等非开源软件所不能了解的软件的各个方面。但在开源的同时,懂得编程语言的人便可以修改软件的代码,从而为他所用,这样虽然可以广泛地提高软件更新的速度,但对于那些对计算机不是很了解的使用者来说,他们不知道软件是如何运行的,当他们打开软件的时候很可能就是被攻击的时候。而对于非开源软件来说,懂不懂编程语言的使用者都不知道软件是如何运行的,或者说不知道软件具体是如何运行的,这样就保证了软件不会被恶意修改并发送给那些容易受到攻击的用户。

3) 操作系统安全性

操作系统也是有安全级别的。美国可信计算机安全评价标准(TCSEC)将计算机系统的安全划分为 D 级、C 级、B 级和 A 级 4 个等级、7 个级别。D 类安全等级只包括 D1 这一个级别,D1 的安全等级最低;C 类安全等级能够提供审慎的保护,并为用户的行动和责任提供审计能力,C 类安全等级可划分为 C1 和 C2 两类;B 类安全等级可分为 B1、B2 和 B3 三类,B 类系统具有强制性保护功能;A 系统的安全级别最高,A 类安全等级只包含 A1 一个安全类别。

4) 功能漏洞

功能上的漏洞可能是编写代码时的疏忽,也可能是未对使用群体做深入的研究,导致软件在使用上容易误操作而导致错误。

5) 软件中涉及的相关不良内容

不良内容包括很多种类,如黄色暴力、假证发票、信用卡诈骗和网络赌博等。

本节中建立模型的指标体系所提及的并非是影响软件信任度的所有因素,而是目前比较受到关注的几个因素。其中,设计中的缺陷包括未对输入过滤、未及时释放内存、非提示弹窗、发送未加密数据和弱口令;源代码方面包括开放源代码和部分开放源代码;操作系统

方面包括验证设计级、强制保护级、自主保护级和最小保护级;功能漏洞方面包括存在高危漏洞和操作烦琐;涉及的相关内容包括涉及黄色暴力、涉及诱导性及欺骗性信息以及涉及政治、宗教和种族的不良信息。

8.2.2 软件信任评价模型框架

1. *层次分析法(AHP)*

层次分析法(Analytic Hierarchy Process,AHP)是在 20 世纪 70 年代中期由美国运筹学家托马斯·塞蒂(T. L. Saaty)正式提出的。AHP 是将一个复杂的多目标决策问题看做一个系统,将与决策有关的元素分解成目标、准则和方案等简单的层次,在此基础之上进行定性和定量分析的决策方法。层次分析法的基本步骤如下。

1) 建立层次结构模型

经过对实际问题进行深入的分析,将与目标有关的各个因素按照不同的类型和不同的属性分成不同的层次,在同一层中的不同因素依据对上层因素的影响不同而分为不同的类别。层级结构的顶端是目标层,是要分析的问题的主要目标。在顶层下面的层次是准则层,准则层的下层为子准则层,各层间按照相似的规则进行划分。

2) 构造比较矩阵

从层次结构模型的第二层开始,从属于(或影响)上一层每个因素的同一层各个因素按成对比较法和一定的比较尺度构造比较矩阵。在构造的比较矩阵中,元素代表因素间相互影响程度,元素的值越大,则元素对于与其比较的元素来说就更加重要。而在以矩阵的对角线相对称的位置则是此元素的倒数,也就是说,重要性是用数值 n 和 $1/n$ 来表示的。

3) 计算权向量并做一致性检验

对于每一个比较矩阵,计算其最大特征根及对应特征向量,利用一致性指标和随机一致性指标计算出的一致性比率做一致性检验。若检验通过,特征向量(归一化后)即为权向量;若没通过,需重新构造或矫正比较矩阵。

4) 计算组合权向量

计算底层目标的组合权向量,按照组合权向量表示的结果进行决策。若层次较多且每层的因素也很多,则需要对组合权向量做组合一致性检验。

2. *ATEMS 模型框架*

软件是用户与计算机硬件之间的接口,用户主要是通过软件与计算机进行交流。由于计算机和网络的广泛应用,软件的数量和覆盖的范围更是大得惊人,所以对于软件的评价和对于软件的使用是分不开的。软件本身又是各种各样、功能各异的,运行在不同的操作系统上,编写软件的语言也是不同的。同时,由于软件的使用者不同,相同的软件可能对于不同的使用者和使用意图起着不同的作用。正是由于软件的这种灵活性和多样性,决定了软件可信性评价模型也应包含多准则、多因素并采用可以考虑到多种方面的评价方法。

本节模型中影响软件信任度的因素均为目前比较受关注的方面。本节将影响软件信任度的因素分为 5 个方面,共 16 个因素。

层次的划分如图 8.1 所示,目标层 A 为软件信任度,准则层 B 分为 B_1、B_2、B_3、B_4 和 B_5。B_1 为设计中的缺陷,B_2 为源代码是否开放,B_3 为操作系统安全性,B_4 为功能漏洞,B_5 为软

件中涉及的相关不良内容。其中，B_1 包括 C_1 未对输入过滤、C_2 未及时释放内存、C_3 非提示弹窗、C_4 发送未加密数据和 C_5 弱口令；B_2 包括 C_6 开放源代码和 C_7 不开放源代码；B_3 包括 C_8 验证设计级、C_9 强制保护级、C_{10} 自主保护级和 C_{11} 最小保护级；B_4 包括 C_{12} 存在高危漏洞和 C_{13} 操作烦琐；B_5 包括 C_{14} 涉及黄色和暴力、C_{15} 涉及诱导性及欺骗性信息和 C_{16} 涉及政治宗教和种族的不良信息。其层级关系如图 8.1 所示。

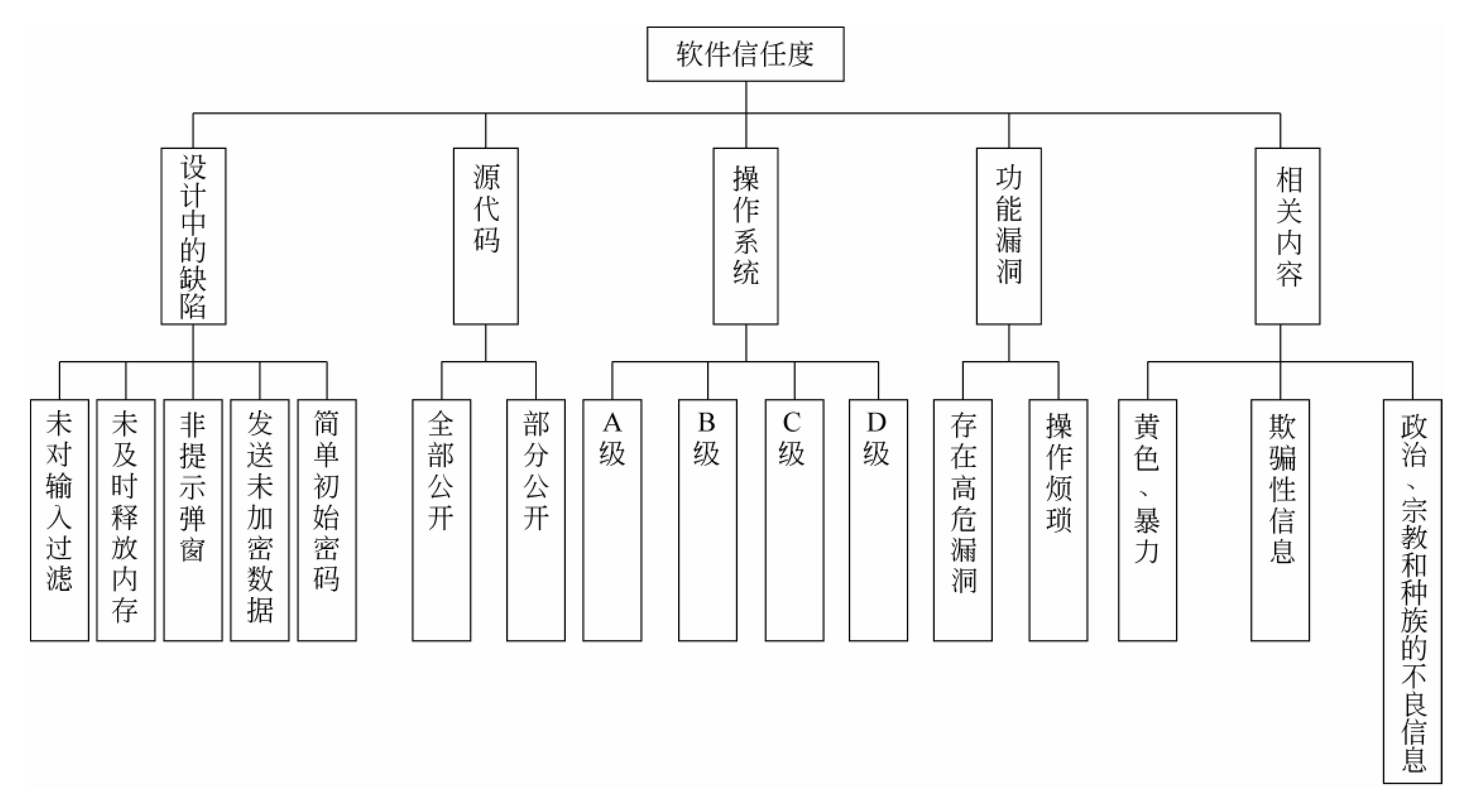


图 8.1　ATEMS 层级模型

3. 构造判断矩阵

由于影响软件信任度的因素可分为不同的层级，并且每层之中各个因素也分为不同的类别，事实上，在每个不同的类别中的不同因素也对软件的信任度有着不同的影响。以操作系统为例，相同的软件在不同的操作系统中运行时有着不同的要求，软件在不同的操作系统中可能有着不同的界面、不同的系统文件调用。由于各层的因素对软件信任度的影响不同，因此需要确定各个因素的重要性顺序作为其权重的参考量。

由于在模型中因素众多，需要可以分出等级且可计算的赋值方法，在 AHP 模型中大多采用 9 度标度法，如表 8.1 所示。

表 8.1　9 度标度法

标度	定　　义	标度	定　　义
1	两因素重要性相同	9	一因素比另一因素极端重要
3	一因素比另一因素稍为重要	2、4、6、8	两相邻判断的中值
5	一因素比另一因素明显重要	倒数	一因素对另一因素的非重要性取其倒数
7	一因素比另一因素强烈重要		

将 9 度标度法应用于 AHP 方法中构造判断矩阵的做法如下。

首先，每层中的不同类别的因素通过 9 度标度法的定义进行重要性比较，得出比较数值。如公开源代码相比于不公开源代码强烈重要，则在得到的 AHP 比较矩阵中，公开源代

码与不公开源代码的重要性位置数值为 7;同理,在不公开源代码与公开源代码的重要性的位置,数值则为 7 的倒数,即 1/7。

以某准则层下的一个类别中的 5 个相对重要性逐级递增的因素为例。假设 5 阶方阵 $\boldsymbol{J}$ 为准则层判断矩阵,则 $\boldsymbol{J}$ 矩阵应为

$$\boldsymbol{J}=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix}=\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ \frac{1}{2} & 1 & 2 & 3 & 4 \\ \frac{1}{3} & \frac{1}{2} & 1 & 2 & 3 \\ \frac{1}{4} & \frac{1}{3} & \frac{1}{2} & 1 & 2 \\ \frac{1}{5} & \frac{1}{4} & \frac{1}{3} & \frac{1}{2} & 1 \end{bmatrix}$$

4. 计算权重值

AHP 模型中,各个因素的权重值的确定是,通过对判断矩阵进行计算得出其最大特征根所对应的特征向量;特征向量归一化后,特征向量中各个数值即为所求权重。特征向量的求解步骤如下。

1)判断矩阵每列归一化

$$\bar{a}_{ij}=\frac{a_{ij}}{\sum_{k=1}^{n}\bar{a}_{kj}}\quad(i,j,k=1,2,\cdots,n)$$

其中,a_{ij} 为准则层 $\mathrm{B}_i(i=1,2,\cdots,n)$ 中子准则层中的影响因素。

2)矩阵的元素按行相加

$$\overline{W}_i=\sum_{j=1}^{n}\bar{a}_{ij}\quad(i,j=1,2,\cdots,n)$$

3)相加后的向量归一化

$$W_i=\frac{\overline{W}_i}{\sum_{j=1}^{n}\overline{W}_j}\quad(i,j=1,2,\cdots,n)$$

5. 一致性检验

判断矩阵是根据决策者的知识和经验得出的,决策者做出的估计可能并不精确。可能是由于可评价的因素过多或评价时受到其他外界影响等,各个因素的重要性之间可能出现矛盾,如 a 比 b 重要,b 比 c 重要,而 c 又比 a 重要。理论上来说,如果 $\boldsymbol{A}$ 是完全一致的成对比较矩阵,应该有 $a_{ij}a_{jk}=a_{ik}$,$1\leqslant i,j,k\leqslant n$。为了避免出现这样的情况,需要对矩阵进行一致性检验并调整矩阵。

判断矩阵的最大特征根由以下公式求出:

$$\lambda_{\max}=\sum_{i=1}^{n}\frac{(AW)_i}{nW_i}$$

其中,

$$W_i = \frac{\overline{W}_i}{\sum_{j=1}^{n} \overline{W}_j} \quad (i,j = 1,2,3,\cdots,n)$$

一致性指标

$$\mathrm{CI} = \frac{\lambda_{\max} - n}{n - 1}$$

其中 n 为判断矩阵的维数。则一致性比率为

$$\mathrm{CR} = \frac{\mathrm{CI}}{\mathrm{RI}}$$

其中平均随机一致性指标 RI 是对于固定的 n，随机构造成对比较矩阵 $\boldsymbol{A}$，其中 a_{ij} 是从 1，2，…，9，1/2，1/3，…，1/9 中随机抽取的。这样得到的 $\boldsymbol{A}$ 是不一致的，取充分大的子样本(大于 500)得到 $\boldsymbol{A}$ 的特征根后求算数平均值得到的。具体数值如表 8.2 所示。

表 8.2　RI 值

维数 n	2	3	4	5	6	7	8	9
RI 值	0	0.58	0.9	1.12	1.240	1.32	1.41	1.45

通过计算 CI 和与判断矩阵维数相对应的 RI，运用上面的公式计算出一致性比率 CR。得到了判断矩阵的 CR 后，便可以判断其一致性，并决定是否对判断矩阵进行调整。当 CR<0.1 时，则认为判断矩阵具有满意的一致性，或者其不一致的程度是在接受的范围内的；否则调整判断矩阵，直到其到达满意的一致性为止。

6. 信任度贡献值和信任度的计算

在本模型中引入 AHP 方法的目的是确定影响软件信任度的因素在计算软件信任度时的权重。为了确定软件的信任度的具体数值，只有各因素的权重是无法计算的。

为此，在本模型中引入各因素的信任度贡献值。顾名思义，信任度的贡献值也就是每层的各个因素对软件信任度的影响程度。贡献度的确定可以用很多方法，如上面的 9 度标度法或 5 度标度法等。为了能够得到比较精确的信任度数值，故在本模型中采用百分制给出贡献值，即各个因素对信任度的贡献度为 0～100，也就是说，60 分以下的因素即被认为对该软件的贡献度不起积极作用。

在得到各个因素的信任度贡献值和权重后，准则层中各个类别的信任度由其包括的子准则层数据计算得出：

$$t_j = \sum_{i=1}^{n} D_i (W_{\boldsymbol{B}_{dj}})_i (d = 1,2,\cdots,l;\ i = 1,2,\cdots,n;\ j = 1,2,3,\cdots,m)$$

在本模型中，$l=2$，$m=5$，$n=2,3,4,5$。

在得出准则层中各个类别的信任度后，目标层的信任度 T 则可由准则层数据计算得出：

$$T = \sum_{j=1}^{p} W_{\boldsymbol{A}_i} t_j \quad (p = 1,2,\cdots,q)$$

本模型中，$i=2$，$q=5$。

8.2.3 实例分析

为了验证本节构建的软件信任评价模型，应用两种日常软件（IM 软件 QQ 和下载软件迅雷）对模型进行实例分析，以检测其可用性和有效性。

1. 数据采集

由前面可知，模型中需要两种数据：信任度的贡献值和判断矩阵。本节通过调查问卷的形式得到这两种数据，方法如下：选取 20 人的样本，对每个人发放两种形式的调查问卷，一种收集模型因素对信任度的贡献值，另一种收集数据来构建判断矩阵。

IM 软件 QQ 的平均贡献度 D_i 如表 8.3 所示。

表 8.3 IM 软件的平均贡献度

子层	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}
D_i	61	85	76	87	62	73	68	91	82	65	74	94	92	87	88	91

下载软件迅雷的平均贡献度 D_i 如表 8.4 所示。

表 8.4 下载软件的平均贡献度

子层	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}
D_i	63	66	85	68	62	74	71	90	67	76	82	90	94	76	88	79

2. 数据处理

由于通过调查问卷得到的受访者对本模型各个因素重要性相比较的数据较多，限于篇幅，这里不将所有得到的数据罗列出来。在以下计算中的判断矩阵均为将数据处理后（简单的加权平均）的结果。

1）对 IM 软件所得数据进行处理

IM 软件 QQ 的判断矩阵 $\boldsymbol{A}_1$、$\boldsymbol{B}_{11}$、$\boldsymbol{B}_{12}$、$\boldsymbol{B}_{13}$、$\boldsymbol{B}_{14}$ 和 $\boldsymbol{B}_{15}$ 分别如下：

$$\boldsymbol{A}_1=\begin{bmatrix}1 & \frac{1}{5} & \frac{1}{4} & \frac{1}{2} & \frac{1}{3}\\ 5 & 1 & 3 & 3 & \frac{1}{2}\\ 4 & \frac{1}{3} & 1 & 2 & \frac{1}{3}\\ 2 & \frac{1}{3} & \frac{1}{2} & 1 & \frac{1}{2}\\ 3 & 2 & 3 & 2 & 1\end{bmatrix}$$

$$W_{A_1}=[0.0667 \quad 0.3041 \quad 0.1681 \quad 0.1178 \quad 0.3433]^T$$

判断矩阵的最大特征根 $\lambda_{max}=5.31$，一致性指标 $CI=0.078$，一致性比率 $CR=\frac{CI}{RI}=0.069<0.1$，矩阵达到一致性标准。

$$B_{11}=\begin{bmatrix}1 & \frac{1}{2} & \frac{1}{4} & 5 & \frac{1}{3}\\ 2 & 1 & \frac{1}{3} & 5 & \frac{1}{4}\\ 4 & 3 & 1 & 5 & \frac{1}{2}\\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & 1 & \frac{1}{5}\\ 3 & 4 & 2 & 5 & 1\end{bmatrix}$$

$$W_{B_{11}}=[0.1254 \quad 0.1525 \quad 0.2681 \quad 0.0488 \quad 0.4052]^T$$

判断矩阵的最大特征根 $\lambda_{max}=5.34$，一致性指标 CI=0.078，一致性比率 $CR=\frac{CI}{RI}=0.076<0.1$，矩阵达到一致性标准。

$$B_{12}=\begin{bmatrix}1 & \frac{1}{7}\\ 7 & 1\end{bmatrix}$$

$$W_{B_{12}}=[0.129 \quad 0.8761]^T$$

当判断矩阵的维数为 2 时，判断矩阵无须验证一致性。

$$B_{13}=\begin{bmatrix}1 & \frac{1}{5} & \frac{1}{4} & \frac{1}{3}\\ 5 & 1 & \frac{1}{3} & \frac{1}{4}\\ 4 & 3 & 1 & \frac{1}{3}\\ 3 & 4 & 3 & 1\end{bmatrix}$$

$$W_{B_{13}}=[0.0897 \quad 0.1807 \quad 0.2563 \quad 0.4733]^T$$

判断矩阵的最大特征根 $\lambda_{max}=5.1$，一致性指标 CI=0.0525，一致性比率 $CR=\frac{CI}{RI}=0.047<0.1$，矩阵达到一致性标准。

$$B_{14}=\begin{bmatrix}1 & \frac{1}{9}\\ 9 & 1\end{bmatrix}$$

$$W_{B_{14}}=[0.0995 \quad 0.9005]^T$$

$$B_{15}=\begin{bmatrix}1 & 2 & 2\\ \frac{1}{2} & 1 & 2\\ \frac{1}{2} & \frac{1}{2} & 1\end{bmatrix}$$

$$W_{B_{15}}=[0.4905 \quad 0.3119 \quad 0.1976]^T$$

判断矩阵的最大特征根 $\lambda_{max}=3.05$，一致性指标 CI=0.025，一致性比率 $CR=\frac{CI}{RI}=0.043<0.1$，矩阵达到一致性标准。

由以上数据计算可知，IM 软件 QQ 的信任度 $T_{IM}=\sum_{j=1}^{p}W_{A_i}t_j=79.33$。

2）对下载软件所得数据进行处理

下载软件迅雷的判断矩阵 $\boldsymbol{A}_2$、$\boldsymbol{B}_{21}$、$\boldsymbol{B}_{22}$、$\boldsymbol{B}_{23}$、$\boldsymbol{B}_{24}$ 和 $\boldsymbol{B}_{25}$ 分别如下：

$$\boldsymbol{A}_2=\begin{bmatrix}1 & \frac{1}{5} & \frac{1}{5} & \frac{1}{3} & \frac{1}{4}\\ 5 & 1 & 3 & 4 & \frac{1}{2}\\ 5 & \frac{1}{3} & 1 & 2 & \frac{1}{3}\\ 3 & \frac{1}{4} & \frac{1}{2} & 1 & \frac{1}{2}\\ 4 & 2 & 3 & 2 & 1\end{bmatrix}$$

$$W_{\boldsymbol{A}_2}=[0.0533\quad 0.3109\quad 0.1675\quad 0.1197\quad 0.3486]^T$$

判断矩阵的最大特征根 $\lambda_{\max}=5.35$，一致性指标 $\mathrm{CI}=0.0875$，一致性比率 $\mathrm{CR}=\frac{\mathrm{CI}}{\mathrm{RI}}=0.078<0.1$，矩阵达到一致性标准。

$$\boldsymbol{B}_{21}=\begin{bmatrix}1 & 5 & 4 & 3 & 2\\ \frac{1}{5} & 1 & \frac{1}{3} & \frac{1}{4} & \frac{1}{5}\\ \frac{1}{4} & 3 & 1 & \frac{1}{3} & \frac{1}{4}\\ \frac{1}{3} & 4 & 3 & 1 & \frac{1}{2}\\ \frac{1}{2} & 5 & 4 & 2 & 1\end{bmatrix}$$

$$W_{\boldsymbol{B}_{21}}=[0.4006\quad 0.0517\quad 0.0942\quad 0.1778\quad 0.2757]^T$$

判断矩阵的最大特征根 $\lambda_{\max}=5.21$，一致性指标 $\mathrm{CI}=0.0525$，一致性比率 $\mathrm{CR}=\frac{\mathrm{CI}}{\mathrm{RI}}=0.047<0.1$，矩阵达到一致性标准。

$$\boldsymbol{B}_{22}=\begin{bmatrix}1 & \frac{1}{6}\\ 6 & 1\end{bmatrix}$$

$$W_{\boldsymbol{B}_{22}}=[0.1441\quad 0.8559]^T$$

$$\boldsymbol{B}_{23}=\begin{bmatrix}1 & \frac{1}{5} & \frac{1}{4} & \frac{1}{3}\\ 5 & 1 & \frac{1}{3} & \frac{1}{4}\\ 4 & 3 & 1 & \frac{1}{3}\\ 3 & 4 & 3 & 1\end{bmatrix}$$

$$W_{\boldsymbol{B}_{23}}=[0.0897\quad 0.1807\quad 0.2563\quad 0.4733]^T$$

判断矩阵的最大特征根 $\lambda_{max}=5.1$，一致性指标 $CI=0.0525$，一致性比率 $CR=\frac{CI}{RI}=0.047<0.1$，矩阵达到一致性标准。

$$B_{24}=\begin{bmatrix}1 & \frac{1}{7}\\ 7 & 1\end{bmatrix}$$

$$W_{B_{24}}=[0.1239\quad 0.8761]^T$$

$$B_{25}=\begin{bmatrix}1 & 3 & 3\\ \frac{1}{3} & 1 & 2\\ \frac{1}{3} & \frac{1}{2} & 1\end{bmatrix}$$

$$W_{B_{25}}=[0.5897\quad 0.2515\quad 0.1588]^T$$

判断矩阵的最大特征根 $\lambda_{max}=3.05$，一致性指标 $CI=0.025$，一致性比率 $CR=\frac{CI}{RI}=0.043<0.1$，矩阵达到一致性标准。

由以上数据计算可知，下载软件迅雷的信任度 $T_{DL}=\sum_{j=1}^{p}W_{A_i}t_j=77.77$。

通过运用建立的评价模型对 IM 软件 QQ 和下载软件迅雷的信任度进行评价的结果可知：79.33>77.77。可知，对于受调查者来说，IM 软件 QQ 的信任度大于下载软件迅雷的信任度。

3. 结论

通过运用建立的评价模型对 IM 软件 QQ 和下载软件迅雷的信任度进行评价的结果可知：79.33>77.77。可知对于受调查者来说，IM 软件 QQ 的信任度大于下载软件迅雷的信任度。与此同时，在参与调查的 20 人中有 13 人认为 IM 软件 QQ 与下载软件迅雷相比具有更高的信任度。本节评价模型的结论与调查的结论相同，从而验证了本节模型的可用性。

8.3　网站信任评价体系

本节提出建立一套网站信任评价体系，来防止网络用户被欺骗或被恶意攻击而选择可信网站的问题。该体系中不仅包含网站的安全制度规范，还包括其对网站信任的评价模型。本节还提出评价体系中网站的安全制度是不可或缺的，并给出了安全制度的基本框架。本节综合网站的类别、网站的漏洞以及网站的安全制度 3 个方面的因素建立评价模型指标体系。通过对两个目前较流行的网站进行分析测试，证明了模型的可用性。

8.3.1　网站信任评价

网站是指根据一定的规则，使用 HTML 等工具制作的用于展示特定内容的相关网页的集合。网站是一种通信工具，通过网站，人们可以发布自己想要公开的资讯或提供相关的

网络服务。许多公司都有自己的网站,他们利用网站来进行宣传、发布产品资讯和招聘信息等。人们可以通过网页浏览器来访问网站,获取自己需要的资讯或者享受网络服务。随着互联网的飞速发展,各种各样的网站应运而生,越来越多的政府部门、商业机构、非营利机构、银行和学校等纷纷将信息和业务“搬”到网上。同时,很多个人也可以制作个人主页,这些通常是用来自我介绍、展现个性的地方,如现在的博客和微博等。这样不仅能让网站的主办单位运用网站为其做宣传并提供各种服务,也方便了需要使用网站的单位和个人。不同的网站含有不同的信息,随着网站数量的激增,网站中包含的信息量也呈爆炸式发展。在搜索引擎中简单地搜索一下就可得到上亿的相关网页。有时在如此海量的信息面前我们显得无所适从,相同或相关的主题可能出现在不同的网站中,但不同的网站间是有区别的。有些网站是正规网站发布正常的消息并且网站本身不存在恶意链接或国家法令禁止的内容;但有些网站上则可能弹出欺骗性的窗体,如仿冒 QQ 的新消息或通知中奖等窗体,还可能有一些涉及黄色、暴力或违禁的信息等。有的网站连接到钓鱼网站,引诱网民进行转账或者交易来骗取钱财,或者存在跨站攻击盗取网民的 Cookie。面对如此众多的网站,如何才能使众多的网络用户不被欺骗或被恶意攻击,而选择信任的网站,就成为首要问题。对众多的网站进行信任评价成为解决该问题的一种方法。本节提出建立一套网站信任评价体系:在制度方面,该体系提出建立统一的网站安全制度标准用于规范安全制度;在模型方面,综合网站的类型、网站安全漏洞以及网站的安全制度建立网站信任评价模型。

8.3.2 影响网站信任度的因素

网站的信任度取决于网站自身的很多因素,网站的信任度有别于软件的信任度。软件一旦编译成功,软件所能提供的功能和信息都是软件编制时设定好的。软件的维护一般都是依靠给软件打补丁或定时更新,并且软件编制成功后都需要有加密或加壳的过程,一般的用户很难了解软件的源代码。而网站对自身的保护则是通过防火墙、IDS 和 IPS 等网络设备或通过服务端对访问控制和输入的字段等进行设置来控制网站的安全性。与此同时,网站的维护需要有一定能力的人员不断地检查,对于网站的管理也要有一定的管理制度。因此,影响网站信任度的因素不仅仅包括网站本身的漏洞,还应包括网站维护、网站管理的制度和规范。

1. 网站的分类

目前互联网上的 IPv4 资源日益耗尽,由此可见互联网上的网站之多,在这海量的网站中,不同的网站提供不尽相同的服务,政府部门网站提供动态信息、办事指南和政策法规等,也可提供网上行政业务申报、办理和相关数据查询等交易类网站为商家提供在线交易的平台等。网站的信任度部分取决于网站提供的服务,如社区网站提供网民在网上进行交流、发布自己观点的平台。其中不乏有人会为了提高自己的关注度而制造一些虚假消息,或为了牟取私利而提供欺骗性消息。

本节对目前的比较流行的网站进行分类,从而为评价网站的信任度提供数据。网站分类及其功能如下。

1) 资讯门户类网站

资讯门户类网站,主要目的是提供信息,是当前最普遍的网站形式之一。此类网站涵盖的工作类型很多,同时信息量就会相对较大,访问的群体也是非常广泛。其功能通常包括检

索功能、论坛板块和留言板等。评价此类网站主要的参考因素包括信息类型、信息发布方式、流程以及信息量的数量级别。如新浪、搜狐和新华网等都属于这类网站。

2）企业品牌类网站

企业品牌网站，顾名思义，目的是为了展示企业综合实力，体现企业 CIS 和品牌理念。网站利用多媒体交互技术和动态网页技术针对客户进行内容建设，从而达到企业形象建设和品牌营销传播的目的。企业品牌网站大致可分为三类：企业形象网站、品牌形象网站和产品形象网站。

3）交易类网站

交易类网站以订单为中心，以实现交易为目的。交易的对象可以是企业或是消费者。此类网站有 3 项基本内容：展示商品、生成订单和执行订单。系统功能一般包括产品管理、订购管理、订单管理、产品推荐、支付管理、收费管理、送发货管理和会员管理等。一些比较大的交易网站还包括积分 ERP 系统、VIP 管理系统、CRM 系统、MIS 系统和商品销售分析系统等。交易类网站可细分为 3 类：B2C 网站、B2B 网站和 C2C 网站。

4）社区网站

社区网站是比较大型的网站，其注册用户的数量相对来说也是较多的。其功能类似于 BBS，比如猫扑和天涯等。

5）办公及政府机构网站

此类网站分为企业办公事务类网站和政府办公类网站。企业办公事务类网站主要包括企业办公事务管理系统、人力资源管理系统、办公成本管理系统和网站管理系统等。政府办公类网站利用外部政务网与内部局域办公网络连接而进行业务的处理。其基本功能有：提供多数据源接口，整合业务系统的数据和发布管理复杂的信息。政府网站既可提供办事指南、政策法规和动态信息等，又可提供网上的行政业务申报、办理以及相关数据查询等。

6）互动游戏网站

互动游戏网站是随着网站服务端的海量数据存储和交互功能的提高而发展起来的新兴网站。此类网站的结构是根据所承载的互动游戏的复杂程度而定的。

7）有偿资讯类网站

有偿资讯类网站以有偿提供资讯为主要业务。这类网站的业务模型一般要求访问者按次、按时间或按量付费。

8）功能性网站

功能性网站近年来兴起的一种应用最广泛的网站，其主要特征是将一个需求非常广泛的功能进行扩展，并开发出与其相应的支撑体系。Google 和百度就是其典型代表。

9）综合类网站

综合类网站提供两种或以上的服务，但网站中不同的服务是由不同的服务商提供的，在其首页中便体现出所提供的服务，如新浪和搜狐。

由于网站的不同，其提供的服务、访问量和影响也不尽相同，网站的可信性也依赖于这些不同的因素。如政府网站是我国各级政府机关履行职能、面向社会提供服务的官方网站，其信任度必定高于社区网站。同时，交易类网站由于其涉及参与用户的经济利益，有国家相关法律法规进行限制，故信任度要高于游戏类网站。若要将网站的分类引入模型，就需要对不同的网站类型进行信任性比较，本节综合网站的影响、涉及的经济利益、访问量以及政策

限制来对不同的类型的网站进行分类。分类的结果如表 8.5 所示。

表 8.5　网站类别可信性次序

网站类别	可信性级别	次序
办公及政府机构网站、交易类网站、有偿资讯类网站	1	1
企业品牌类网站、资讯门户类网站、功能性网站	2	2
互动游戏网站、社区网站	3	3

2. 网站的安全漏洞

随着计算机技术和网络技术的飞速发展,以及网络黑客软件的易操作性和易获得性的提高,使得网站暴露出越来越多的漏洞。这些漏洞轻则影响用户的使用和网站的管理,重则会导致用户敏感数据的丢失和服务器的崩溃,从而导致用户利益受损和网站运营商经济利益的损失。由于这些漏洞均会对用户和网站运营商造成损害,因此本节模型将网站的漏洞作为影响网站信任评价模型的因素。

参考 OWASP 的 webgoat,网站的漏洞大致可分为如下 17 种。

1) 访问控制漏洞(access control flaws)

访问控制漏洞包括基于访问控制的旁路(bypass a path based access control scheme)和远程用户登录(remote admin access)。基于访问控制的旁路是运用../这个跳转字符来跳转到服务器的上层目录中,可能造成未授权用户进行越权浏览;远程用户登录是运用与 admin 字段相同的字段或运用字典来猜测网站的后台管理员登录界面。

2) AJAX 安全性(AJAX Security)

此类漏洞包括基于 DOM 的跨站脚本攻击或注入(DOM-based cross-site scripting or injection)、无声交易攻击(silent transactions attacks)、eval 的不安全用法和不安全的客户端存储(insecure client storage)。

基于 DOM 的跨站脚本攻击,即在 DOM 中插入恶意代码,如,可将代码填入文本输入框中;而基于 DOM 的注入则是在 DOM、XML 和 JSON 中注入代码。无声交易攻击是指交易中如果网页中的应用允许简单的 URL 提交,在 AJAX 中后果会更加严重,即在交易的过程中和交易后,用户收不到任何反馈页面。PHP 中 eval 的用法可以允许用户输入代码,一旦用户输入的是恶意攻击代码,如 123');alert(document.cookie);(',则可进行脚本攻击。存储在客户端的数据可以被软件(如 Firebug)轻易地得到。

3) 认证漏洞(authentication flaws)

认证漏洞是比较简单的一类漏洞,其中包括密码的强度(password strength)、丢失密码的取回过程漏洞、授权码不严密和多重登录。

其中密码强度是众所周知的漏洞,最近的 Oracle Pending 正是由于管理员只用 4 位数字作为密码而造成巨大的经济损失。在找回密码的过程中可以通过拦截软件修改找回密码的使用者,从而获得其他用户的密码。授权码不严密是指在用户得到授权之后,服务器端通过一定的算法计算出授权码对用户进行授权,而简单的算法会使授权码被破解,从而达到提权的效果。多重登录指通过拦截软件进行多重登录从而得到其他用户的信息的过程。

4）缓冲溢出（buffer overflows）

缓冲溢出是通过向缓存中发送超出其所能承载量的数据而导致缓冲溢出至其他内存区，而造成任意代码的执行。

5）源代码质量（code quality）

开发者往往在页面中留有对自己的代码相关的注释等敏感信息，使得恶意用户可通过浏览其源代码而得到攻击的线索。

6）同时登录（concurrency）

同时登录漏洞一般是由于存在线程安全问题（thread safety problems）和不同的浏览器不兼容问题。线程不安全可能造成不同用户同时登录时，用户可以浏览其他用户的信息。而同一用户使用两个不同的浏览器时，可能造成网页中表单数据的变化。

7）跨站脚本（cross-site scripting XSS）

跨站脚本攻击是目前最流行的攻击方法之一，通过跨站脚本攻击，黑客可以实现很多目标，如钓鱼和转账等。跨站脚本攻击包含很多类型，其中有钓鱼跨站攻击（phishing with XSS）、存储式跨站攻击（stored XSS attacks）、反射跨站攻击（reflected XSS attacks）、跨站请求伪造（Cross Site Request Forgery，CSRF）、通过跨站请求伪造来绕过认证（CSRF prompt by-pass）、通过跨站请求伪造 Token 绕过以及跨站追踪攻击（cross site tracing XST attacks）。

钓鱼跨站攻击是通过跨站脚本使浏览者浏览的链接弹出钓鱼表单或页面用以骗取用户的银行卡、信用卡或其他账户的用户名和密码。存储式跨站攻击通过将跨站脚本插入到服务器端的表单或页面中使浏览该表单或页面的用户受到攻击。此种攻击的效果非常厉害，如果存储式跨站发生在知名人士的博客或微博中，所有浏览的人都会受到攻击。反射跨站攻击是通过向受害者发送一段嵌有脚本代码的链接，当受害者点击链接的时候，跨站攻击便成功了。跨站请求伪造是指用户在某个会话中对某个 CGI 进行操作时身份认证信息被劫持或篡改，使得攻击者得到受攻击者的权限或认证的一种攻击。通常的做法是通过脚本代码使受害者载入一个包含链接的图片，当受害者的浏览器试图解析图片链接时，即实现图片链接中的功能，如转账等，然而发送转账请求时用的是受害者的 Cookie，也就是以受害者的身份进行的。还可通过跨站请求伪造得到 Cookie 或 Token，从而达到绕过认证来实现恶意目标。跨站追踪攻击是融合跨站脚本和 HTTP Trace 功能的一种攻击。

8）拒绝服务攻击（Denial of Service）

拒绝服务攻击即向受害主机发送超过其接受量的请求，使被攻击方资源耗尽（CPU 满负荷或内存不足），从而导致其无法提供服务的攻击。

9）错误不当处理（improper error handling）

在网页中出现错误在所难免，但如果处理不当，则可能导致暴露出敏感信息的结果。如 JSP 服务器中，Fail Open Authentication Scheme 在 Java 代码中的异常处理程序执行身份验证成功的 catch 块出现异常时（可能因为存在一个 NullPointer 异常），如果处理不当，则可能暴露密码参数。

10）注入漏洞（injection flaws）

注入是目前漏洞中危险性最大的漏洞之一，由于注入攻击简单易用，并且手法灵活，其攻击形式看起来与 Web 页面访问相同，从而使得注入攻击很难被防火墙所发现，再通过巧

妙构造的注入语句，可以实现窃取资料、执行命令等多种用途。本节主要关注 SQL 注入。

注入攻击的类别很多，主要包括命令注入(command injection)、数字型注入(numeric SQL injection)、日志欺骗(log spoofing)、XPATH 注入(XPATH injection)、字符型注入(string SQL injection)、修改、添加或删除数据(modify, add or delete data with SQL injection)和数据库后门(aatabase backdoors)。

SQL 命令注入通过注入来实现命令执行，最简单的如上面所提到的用../来跳转目录。数字型注入就是存在注入的字段是数字类型的，如连接类型为 http://www.asdf.com?id=1，可在数字类型的 1 后面插入 SQL 语句。字符类型的注入类似于数字型的注入，只是可注入的字段类型为字符型。日志欺骗即在查询语句中插入 CRLF 换行和要插入的虚假日志用以迷惑管理员，既可以通过 SQL 注入来插入查询条件，也可以通过插入如 modify、add 和 delete 来修改、添加和删除数据库中的信息。数据库后门则是通过在数据库中插入一段代码来实现的后门功能。

11) 不安全通信(insecure communication)

不安全通信即通信时发送明文或弱的加密法加密的敏感信息。这样的报文一旦被拦截，被解密的可能非常大。

12) 不安全配置(insecure configuration)

未启用安全的配置或配置文件的文件名未做修改，如比较敏感的 conn.asp，这样很容易使配置文件被攻击者获得而造成损失。

13) 不安全存储(insecure storage)

对于存储在数据库中的敏感信息，如管理员账号和密码，未加密或用弱的加密法加密，攻击者通过暴库轻而易举地得到管理员信息。

14) 恶意执行(malicious execution)

恶意执行是指网站中允许上传文件的地方如未做限制而导致攻击者上传恶意软件或木马。如上传照片的表单通过修改表单的限制值可能使得木马上传成功。

15) 参数篡改(parameter tampering)

由于网页代码编写不严谨导致网页参数遭到篡改。一般包括绕过 HTML 域限制(bypass HTML field restrictions)、隐藏域发现(exploit hidden fields)和绕过客户端 JavaScript 认证(bypass client side JavaScript validation)。

绕过 HTML 域限制即在网页代码中对某些字段的限制只在页面中的某些属性中做限制，通过修改限制值(如将 false 改成 true)就可修改网页中发送的参数。隐藏域发现是指在网页中通过设置隐藏字段来限制一些属性或提示一定的信息，只要通过拦截软件或直接阅读源代码即可发现隐藏字段，获得敏感信息。绕过客户端 JavaScript 认证是通过修改网页中发送请求的参数来绕过 JavaScript 客户端的认证。

16) 会话管理漏洞(session management flaws)

此类漏洞的发生是由于未对会话过程做周密的考虑，从而使得认证 Cookie 被仿造或会话被劫持(hijack a session)和篡改(session fixation)。由于未考虑到复杂性和随机性的安全问题，从而造成会话被劫持，造成中间人攻击，或者攻击者得到用户的 sessionID 从而绕过认证。

17）网络服务漏洞（Web Services flaws）

本节提到的网络服务漏洞指网络服务中运用 SOAP 发送请求，这些请求的发送是为了执行某种由网络服务定义语言（WSDL）定义的功能。通过在 URL 后面简单地添加"?WSDL"就可以浏览 WSDL 文件，导致重要信息的丢失。注入也可能发生在此种情况中，如果输入完全由网络服务的前端限制，那么注入可能损坏网页界面发送的 XML。

由于拒绝服务攻击是由于网络协议本身的安全缺陷造成的，所以在本节中采用其余 16 种漏洞作为评价因素。由于以上每种漏洞均可能造成轻度的损失或严重的损失，这给评价工作带来了困难。因此本节采用 OWASP 在 2012 年提出的 2010 年十大应用程序安全风险来对以上列出的网站漏洞进行评价打分，用以确定漏洞的危险级别。这样不仅可使模型的数据有据可依，还可以保证模型因素的时效性。确定漏洞的危险级别后，可依此构造比较矩阵，从而计算一个漏洞在模型中所占的权重值。

根据 OWASP 的 2010 年十大应用程序安全风险，风险程度由高到低依次为注入、跨站、失效的身份认证和会话管理、不安全的直接对象引用、CSRF、安全配置错误、不安全的加密存储、没有限制 URL 访问、传输层保护不足以及未验证的重定向和转发。

风险名称、存在风险的漏洞编号以及风险分值如表 8.6 所示。

表 8.6　漏洞风险评分

风 险 名 称	风险分值	存在此风险的漏洞编号
注入	10	10
跨站	9	7
失效的身份认证和会话管理	8	1,3,6,16
不安全的直接对象引用	7	5,9
CSRF	6	7
安全配置错误	5	2,4,14
不安全的加密存储	4	12,13
没有限制的 URL 访问	3	17
传输层保护不足	2	11
未验证的重定向和转发	1	15

通过表 8.6 对漏洞进行风险评价打分，评价后的漏洞如表 8.7 所示。

表 8.7　漏洞危险性排序

分值	15	10	8	7	5	4	3	2	1
编号	7	10	1,3,6,16	5,9	2,4,14	12,13	17	11	15
次序	9	8	7	6	5	4	3	2	1

3. 网站的安全制度

网站的信任度不仅取决于网站硬件设备的精良或网站代码的质量，还要依靠体系化的、完善的制度。一个合理的、有效的、适宜的制度能规范员工的行为，提高员工的工作效率和质量，从而提高内部的安全性。正如我们所知，即使网站备有宽敞的机房和先进的设备，也无法抵挡网站内部维护人员的误操作或违规操作。因此网站的安全制度也是网站信任度的决定性因素，而此项因素却常常被忽略。因此，本节提出将网站的安全制度引入网站信任评

价模型中。

目前还没有统一的、系统的网站安全制度，可以参考的多是学校和企业的网站安全制度。安全制度是依照法律、法令和政策制定的具有约束力的、保证正常工作开展的应用文档。其编制要考虑到自身的各种因素、依据法律法规并且要在制度施行期间不断地修改和完善。同时网站安全制度也有其适用群体，安全性要求较高的网站运营商应制定更高的标准。网站安全制度包括的方面广，各种规定细致，并且要根据不同的网站制定相应的安全制度。因此本节仅简单拟定网站安全制度框架，用以在宏观上将网站的安全制度引入评价模型中。综合各高校和企业的网站安全制度，本节提出如图 8.2 所示的网络安全制度框架。

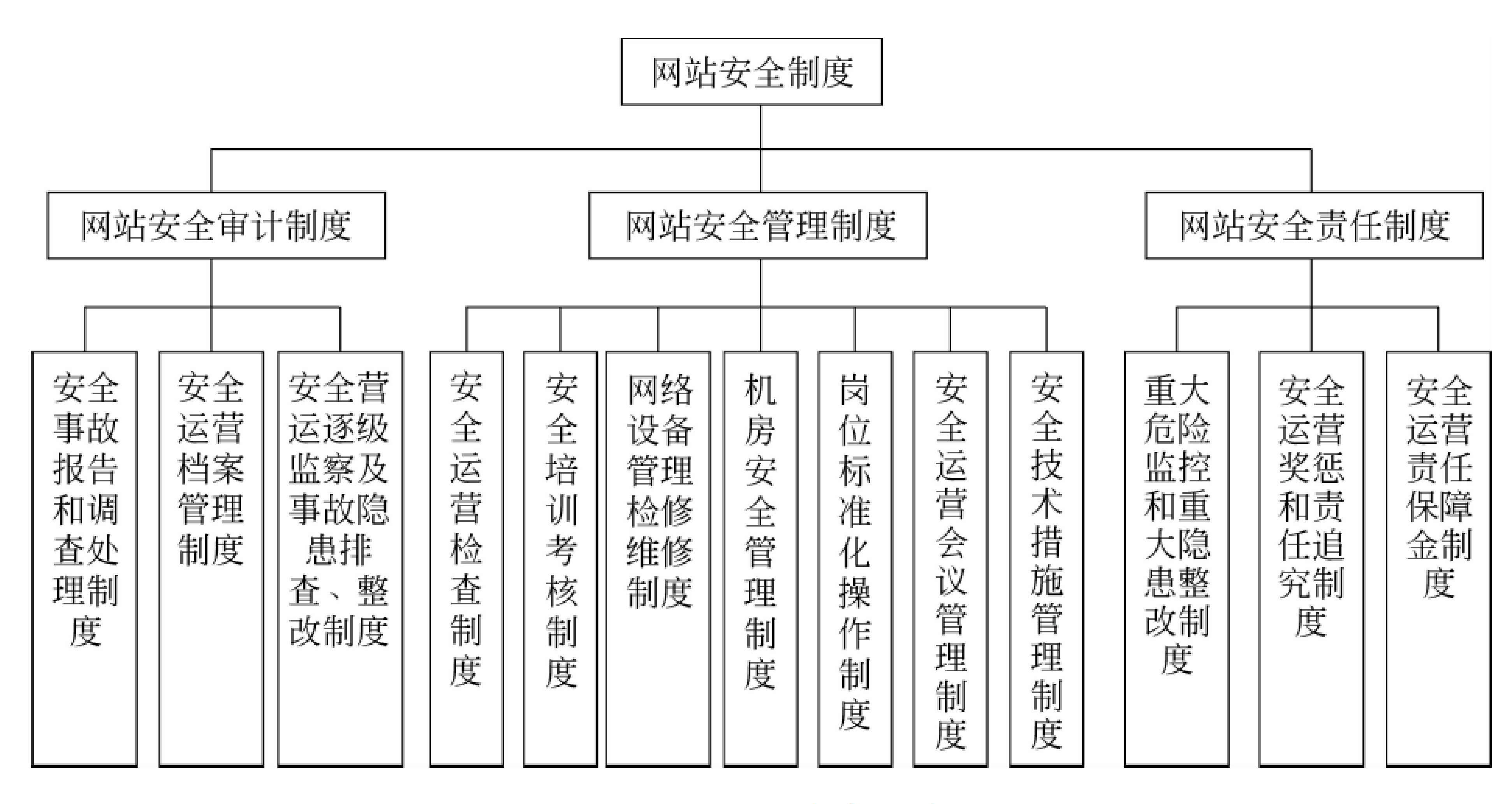


图 8.2　网站安全制度

8.3.3　ATEMW 模型框架及模型检验

网站为网络用户提供资讯和服务，由于不同类型的网站、存在不同漏洞的网站和不同安全制度的网站有着不同的信任度，为了更好地评价网站的信任度，本节将网站的类型、网站的各种漏洞以及网站的安全制度引入评价模型中，建立一种基于 AHP 的网站信任评价模型(简称 ATEMW)。

1. ATEMW 模型框架

本节模型中影响网站信任度的因素分为 3 个方面：网站的类型、网站的安全漏洞以及网站的安全制度。目标层和准则层的结构如图 8.3 所示。

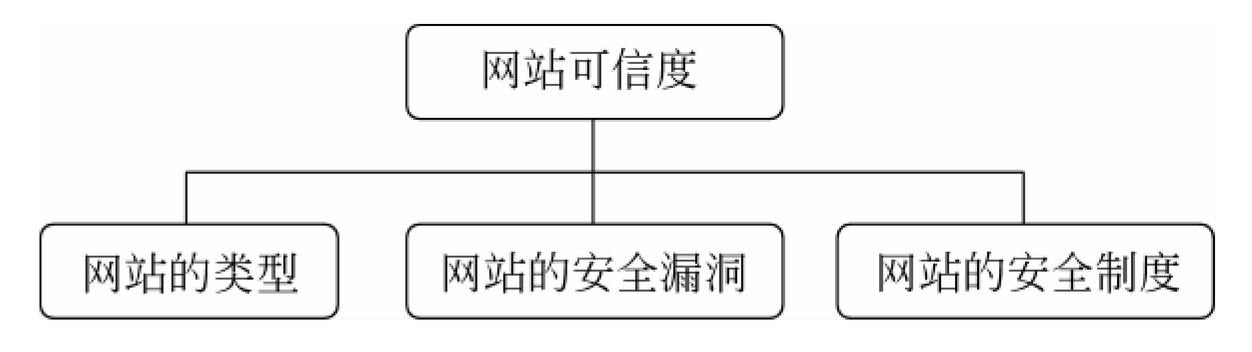


图 8.3　ATEMW 模型框架

准则层中的 3 个方面包含各自的子准则层。网站的类型分为办公及政府机构网站、交易类网站、有偿资讯类网站、企业品牌类网站、资讯门户类网站、功能性网站、互动游戏网站和社区网站。网站分类结构如图 8.4 所示。

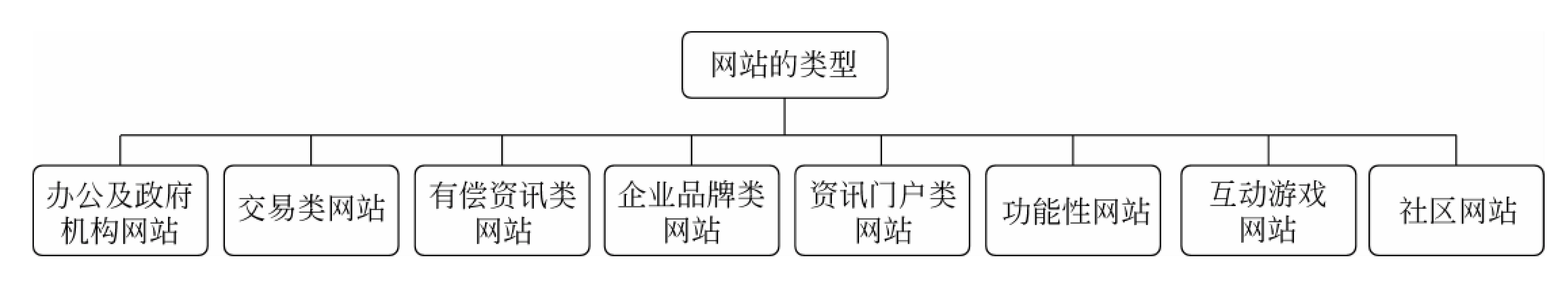


图 8.4 网站分类结构

根据 8.3.2 节所述，网站的漏洞包括访问控制漏洞、AJAX 安全性、认证漏洞、代码质量、缓冲溢出、同时登录、跨站脚本、错误不当处理、注入漏洞、不安全通信、不安全配置、不安全存储、恶意执行、参数篡改、会话管理漏洞和网络服务漏洞。网站漏洞结构如图 8.5 所示。

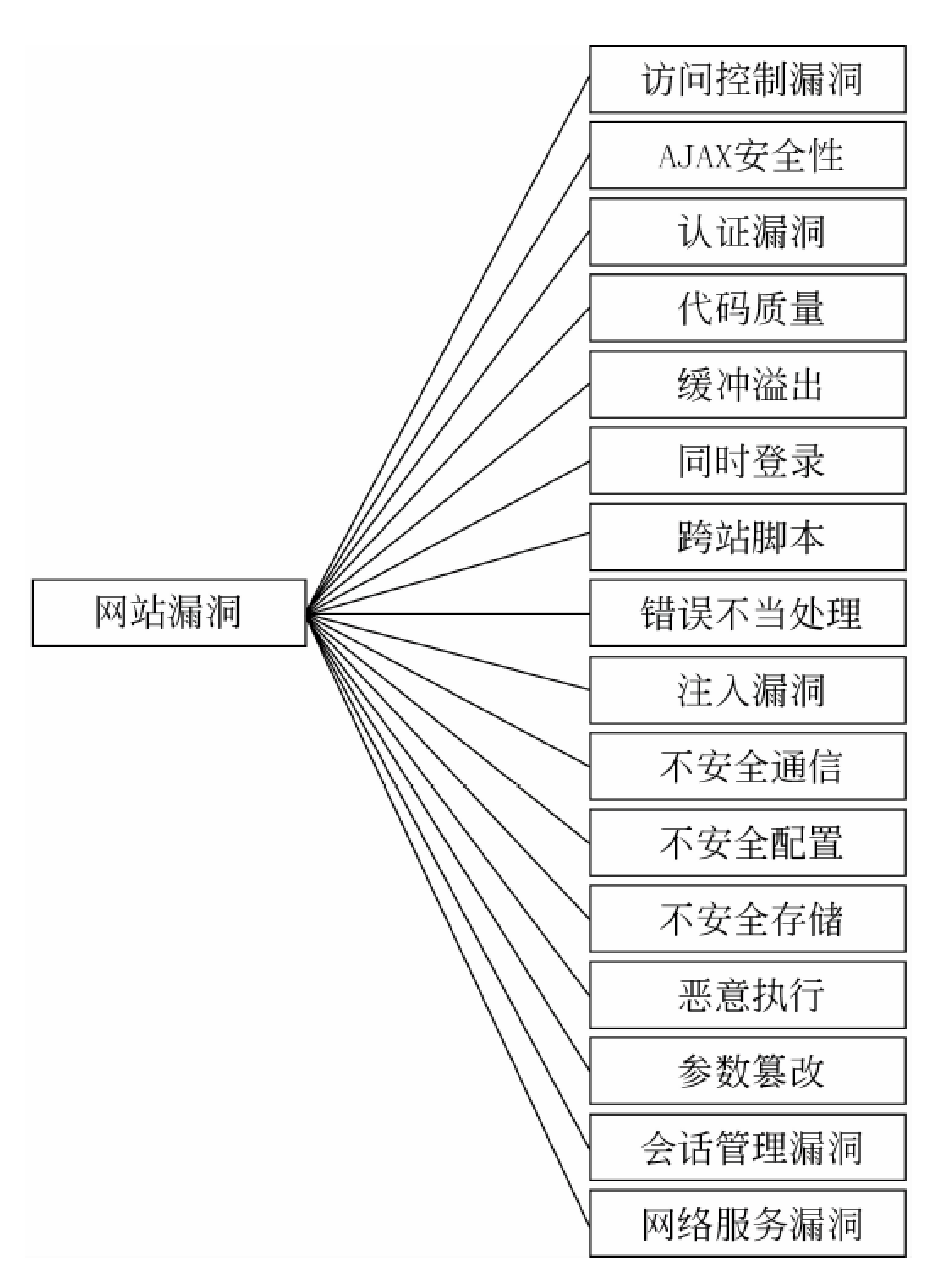


图 8.5 网站漏洞结构

网站安全制度包括的内容如 8.3.2 节所述，其图形可参考图 8.2。

2. ATEMW 建模及检验

网站信任评价不同于软件的信任评价，但与其类似。不同的人使用相同的软件来实现不同的目的，一般的软件可以由客户进行配置，由此实现不同客户所需的功能。但网站绝大多数的配置是在服务端进行的，极少数的客户端配置也只是界面的变化。由本节前面的内容可知，网站的信任度不仅仅取决于网站的类型和网站的漏洞，还取决于网站的安全制度。

1）构造判断矩阵

要运用 AHP 方法对网站的信任度进行评价，就要对不同因素确定其重要性顺序，用来作为计算因素的权重。表 8.5 和表 8.7 已经将网站的类型和网站的漏洞进行了重要性排序。本节中同样使用 9 度标度法（参见表 8.1）。由于在本节的前面已经将网站的类型和网

站的漏洞的重要性顺序排列完成，下面的工作就是用重要性排序来构造比较矩阵。构造比较矩阵的方法同 8.2 节。

由于网站的信任度取决于 3 种因素：网站的类别、网站的漏洞以及网站的安全制度。这 3 者属于不同的方面并且对网站的信任度都有着重要的影响作用。经过对一定数量的受访者进行调查，绝大多数的人认为以上三者对网站的信任度有着相同的作用，三者并不存在重要性的次序问题。因此，准则层的比较矩阵 $\boldsymbol{Z}$ 为

$$\boldsymbol{Z}=\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

软件类别的信任度标度由其可行性级别决定，为了计算简单，在本节中采用标度的最后 3 个标度来确定。网站类别标度确定如表 8.8 所示。

表 8.8　网站类别标度确定

网 站 类 别	可信性级别	标度
办公及政府机构网站、交易类网站、有偿资讯类网站	1	3
企业品牌类网站、资讯门户类网站、功能性网站	2	2
互动游戏网站、社区网站	3	1

网站类别的各种因素序列为办公及政府机构网站、交易类网站、有偿资讯类网站、企业品牌类网站、资讯门户类网站、功能性网站、互动游戏网站和社区网站。

因此可知网站类别的比较矩阵 $\boldsymbol{C}_1$ 为

$$\boldsymbol{C}_1=\begin{bmatrix}1 & 1 & 1 & 2 & 2 & 2 & 3 & 3\\ 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3\\ 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3\\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 & 1 & 1 & 2 & 2\\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 & 1 & 1 & 2 & 2\\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 & 1 & 1 & 2 & 2\\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 & 1\\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 & 1\end{bmatrix}$$

由 8.3.2 节的结果可知，通过 OWASP 公布的十大风险对漏洞进行打分和排序，这样不仅简化了指标体系保证了指标的时效性。根据 OWASP 发布的 2010 年十大应用程序安全风险，风险程度由高到低依次为注入、跨站、失效的身份认证和会话管理、不安全的直接对象引用、CSRF、安全配置错误、不安全的加密存储、没有限制 URL 访问、传输层保护不足以及未验证的重定向和转发。16 种漏洞由以上十大风险打分并排序以适合 AHP 方法，如表 8.7 得出顺序。由于有两大风险出现在同一漏洞中，故网站漏洞的相对比较矩阵应为 9×9 矩阵。因此，网站漏洞的相对比较矩阵 $\boldsymbol{C}_2$ 为

$$
C_2 = \begin{bmatrix}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\frac{1}{2} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\frac{1}{3} & \frac{1}{2} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\frac{1}{4} & \frac{1}{3} & \frac{1}{2} & 1 & 2 & 3 & 4 & 5 & 6 \\
\frac{1}{5} & \frac{1}{4} & \frac{1}{3} & \frac{1}{2} & 1 & 2 & 3 & 4 & 5 \\
\frac{1}{6} & \frac{1}{5} & \frac{1}{4} & \frac{1}{3} & \frac{1}{2} & 1 & 2 & 3 & 4 \\
\frac{1}{7} & \frac{1}{6} & \frac{1}{5} & \frac{1}{4} & \frac{1}{3} & \frac{1}{2} & 1 & 2 & 3 \\
\frac{1}{8} & \frac{1}{7} & \frac{1}{6} & \frac{1}{5} & \frac{1}{4} & \frac{1}{3} & \frac{1}{2} & 1 & 2 \\
\frac{1}{9} & \frac{1}{8} & \frac{1}{7} & \frac{1}{6} & \frac{1}{5} & \frac{1}{4} & \frac{1}{3} & \frac{1}{2} & 1
\end{bmatrix}
$$

由 8.3.2 节可知，网站的信任度取决于网站的安全制度。但是网站的安全制度是一个整体，并且制度的不同规定之间没有轻重之分，对制度的评价只能是完善与否，是否有应用价值，是否能够使使用制度的主体达到预期的效果。对于制度的评价取决于制度所产生的影响，即是否可以使应用制度的主体达到预期的效果，但要达到预期效果需要一个过程，在此过程之中发现制度中的不足并对其进行修改和完善。因此，对于制度的评价并非一个时点值、有一套体系，而是一个制度完善的过程。所以在本节模型中，对于制度的评价只给安全制度一个权重，只初步判断其是否有相对完善的制度存在。

2）计算权重值并检验

通过对比较矩阵进行处理，从而得到矩阵的最大特征根对应的特征向量，对特征向量进行归一化处理后的向量的每个元素就是要求的权重值。特征向量的求解步骤如下。

（1）判断矩阵每列归一化：

$$\overline{a_{ij}} = \frac{a_{ij}}{\sum_{k=1}^{n} \overline{a_{kj}}} \quad (i,j = 1,2,\cdots,n)$$

其中，a_{ij} 为准则层 $B_i(i=1,2,\cdots,n)$中子准则层中的影响因素。

（2）矩阵的元素按行相加：

$$\overline{W_i} = \sum_{j=1}^{n} \overline{a_{ij}} \quad (i,j = 1,2,\cdots,n)$$

（3）相加后的向量归一化：

$$W_i = \frac{\overline{W_i}}{\sum_{j=1}^{n} \overline{W_j}} \quad (i,j = 1,2,\cdots,n)$$

一致性验证运用 AHP 方法通用的验证方法，即求得一致性比率 $CR=\frac{CI}{RI}$。当 $CR<0.1$ 时，则认为判断矩阵具有满意的一致性，或者其不一致的程度是在可接受的范围内；否则调

整判断矩阵直到其达到满意的一致性为止。

分别对比较矩阵 $\boldsymbol{Z}$、$\boldsymbol{C}_1$ 和 $\boldsymbol{C}_2$ 进行求解。

$$\boldsymbol{Z}=\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$$W_Z=[0.33\quad 0.33\quad 0.33]$$

判断矩阵的最大特征根 $\lambda_{max}=1$，一致性指标 $CI<0$，$CR=\dfrac{CI}{RI}<0$。矩阵达到一致性标准。

$$\boldsymbol{C}_1=\begin{bmatrix}1&1&1&2&2&2&3&3\\1&1&1&2&2&2&3&3\\1&1&1&2&2&2&3&3\\\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&1&1&1&2&2\\\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&1&1&1&2&2\\\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&1&1&1&2&2\\\frac{1}{3}&\frac{1}{3}&\frac{1}{3}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&1&1\\\frac{1}{3}&\frac{1}{3}&\frac{1}{3}&\frac{1}{2}&\frac{1}{2}&\frac{1}{2}&1&1\end{bmatrix}$$

$$W_{C_1}=[0.1918\quad 0.1918\quad 0.1918\quad 0.1032\quad 0.1032\quad 0.1032\quad 0.0574\quad 0.0574]$$

判断矩阵的最大特征根 $\lambda_{max}=8.0124$，一致性指标 $CI=0.0018$，$CR=\dfrac{CI}{RI}=0.0013<0.1$。矩阵达到一致性标准。

$$\boldsymbol{C}_2=\begin{bmatrix}1&2&3&4&5&6&7&8&9\\\frac{1}{2}&1&2&3&4&5&6&7&8\\\frac{1}{3}&\frac{1}{2}&1&2&3&4&5&6&7\\\frac{1}{4}&\frac{1}{3}&\frac{1}{2}&1&2&3&4&5&6\\\frac{1}{5}&\frac{1}{4}&\frac{1}{3}&\frac{1}{2}&1&2&3&4&5\\\frac{1}{6}&\frac{1}{5}&\frac{1}{4}&\frac{1}{3}&\frac{1}{2}&1&2&3&4\\\frac{1}{7}&\frac{1}{6}&\frac{1}{5}&\frac{1}{4}&\frac{1}{3}&\frac{1}{2}&1&2&3\\\frac{1}{8}&\frac{1}{7}&\frac{1}{6}&\frac{1}{5}&\frac{1}{4}&\frac{1}{3}&\frac{1}{2}&1&2\\\frac{1}{9}&\frac{1}{8}&\frac{1}{7}&\frac{1}{6}&\frac{1}{5}&\frac{1}{4}&\frac{1}{3}&\frac{1}{2}&1\end{bmatrix}$$

$$W_{C_2}=[0.3070\quad 0.2182\quad 0.1543\quad 0.1089\quad 0.0764\quad 0.0533\quad 0.0370\quad 0.0259\quad 0.0189]$$

判断矩阵的最大特征根 $\lambda_{max}=9.4038$，一致性指标 CI＝0.0505，$CR=\frac{CI}{RI}=0.0384<0.1$。矩阵达到一致性标准。

3）信任度的计算

在得到各个因素的信任度贡献值和权重后，准则层中各个类别的信任度由其包括的子准则层数据计算得出：

$$t_j=\sum_{i=1}^{n}G_i(W_{C_{di}})_i \quad (d=1,2,\cdots,l;\ i=1,2,\cdots,n;\ j=1,2,\cdots,m)$$

在本模型中 $l=3, m=3, n=3,8,9$。

在得出准则层中各个类别的信任度后，目标层的信任度 $\boldsymbol{T}$ 则可由准则层数据计算得出

$$T=\sum_{j=1}^{p}W_{Z_i}t_j \quad (p=1,2,\cdots,q)$$

本模型中 $i=2, q=3$。

8.3.4　实例分析

由于本节中的模型是对于网站信任度的评价模型，故需要对相应的网站进行数据采集。网站的类别和网站的制度可以通过从各个方面搜集信息而得到相关的数据进行评价。但在检验网站漏洞的时候必然要加上测试字段来检验漏洞是否存在，与此同时就会对网站造成不同的影响。如存储式跨站漏洞一旦成功，难免会有存在漏洞的网页被浏览的可能，从而造成访问者看到测试漏洞页面。一旦这样的情况发生，不仅会使访问者对该网站产生反感，也会影响网站的维护工作。因此，在本节中通过虚拟机搭建两个不同的网站用于测试。考虑到网站的服务器可能不尽相同，比较经典的服务器包括 ASP 服务器、PHP 服务器和 JSP 服务器，因此本节中搭建两种服务器（ASP 服务器和 PHP 服务器）用于建立站点以进行测试，在 ASP 服务器中建立动力空间，在 PHP 服务器中建立一个 PHPWind 论坛。

1. 搭建网站并获取数据

首先，我们搭建比较流行的 ASP 服务器，由于本节中建立的站点规模不是很大，同时 Access 数据库具有安装简便、占用的空间小等优点。因此，本节采用的是 ASP＋Access 技术。运用 IIS 搭建动力空间，网站名称为 ATEMWinIIS，如图 8.6 所示。

运用虚拟机的端口映射技术将端口映射到实体机，运用扫描软件对其进行扫描，得到漏洞数据，如图 8.7 所示。

经过验证可知，实际漏洞为 7 个跨站脚本编制、5 个跨站请求伪造和 10 个应用的程序错误。由表 8.7 可知漏洞的重要性可知，危险性排序为跨站脚本攻击；注入漏洞；访问控制漏洞、认证漏洞、同时登录、会话管理漏洞；源代码质量、错误不当处理；AJAX 安全性、缓冲溢出、恶意执行；不安全配置、不安全存储；网络服务漏洞；不安全通信；参数篡改。因此，本节按照漏洞的危险性计算网站不同漏洞的贡献值。由于存在漏洞属于负面影响，故本节中将漏洞的贡献值设置为负值。其分值分别是－4.5、－4、－3.5、－3、－2.5、－2、－1.5、－1、－0.5，如表 8.9 所示。



图 8.6 动力空间

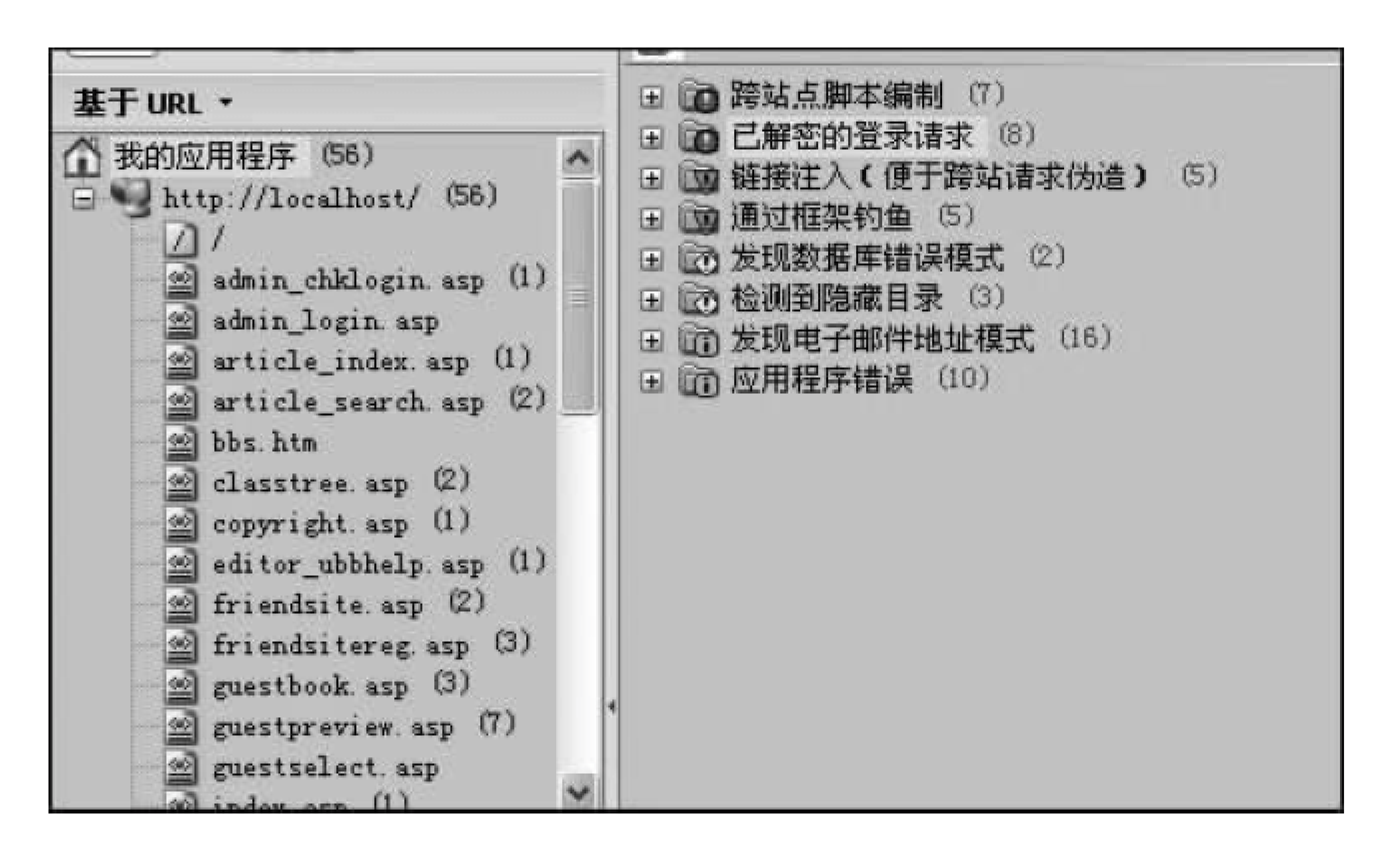


图 8.7 动力空间漏洞

表 8.9 漏洞及贡献值

漏洞编号	7	10	1,3,6,16	5,9	2,4,14	12,13	17	11	15
贡献值 G	−4.5	−4	−3.5	−3	−2.5	−2	−1.5	−1	−0.5

不同的网站,其漏洞的类型和数量也不尽相同。在本节的模型中各种漏洞贡献值 G 的计算方法是由漏洞的数量 n 和其对应的分数 s 的乘积:

$$G = n \times s$$

经过检验可知跨站脚本攻击成功;隐藏目录均返回 403 错误;暴露出的应用错误信息可以了解服务器的相关配置。因此可知各漏洞的贡献值如表 8.10 所示。

表 8.10 漏洞贡献值加总表

漏洞	跨站脚本	错误不当处理	其他
贡献值 G	−31.5	−30	0

其次,我们搭建 PHP 服务器,采用的是 Windows 2003+Apache+PHP+MySQL 的组

合。安装的软件分别为 httpd-2.2.21-win32-x86-no_ssl.msi(Apache)、php-5.2.17-Win32-VC6-x86.msi(PHP)、mysql-essential-5.5.17-m3-win32.msi(MySQL)和 phpwind_GBK_8.7.zip。

通过对 Apache Software Foundation\Apache2.2\conf\httpd.conf 文件中添加以下代码：

```
LoadModule php5_module "C:/Program Files/PHP/php5apache2_2.dll"
PHPIniDir "C:/Program Files/PHP"
AddType application/x-httpd-php .php
```

对 PHP 进行配置，使其与 MySQL 连接，添加后重启 Apache 服务器，安装完成。搭建服务器完成后，创建 PHPWind 论坛的信息如图 8.8 所示。



图 8.8　PHPwind 论坛安装

对虚拟机中新建的 PHPWind 论坛通过扫描软件进行扫描，得到如图 8.9 所示的漏洞。

但经过验证，Flash 参数的允许脚本执行设置为 always 真正存在；robots.txt 未暴露出站点结构；目录列表中的 url 均为无效页面，均返回 403 错误；直接访问管理页面均返回 forbidden。因此可知各漏洞的贡献值如表 8.11 所示。

表 8.11　漏洞贡献值加总表

漏洞	不安全配置	其他
贡献值 G	−8	0

由于不同类别的网站有着不同的贡献值，因此改造表 8.5 得到不同类别的网站的贡献值，如表 8.12 所示。

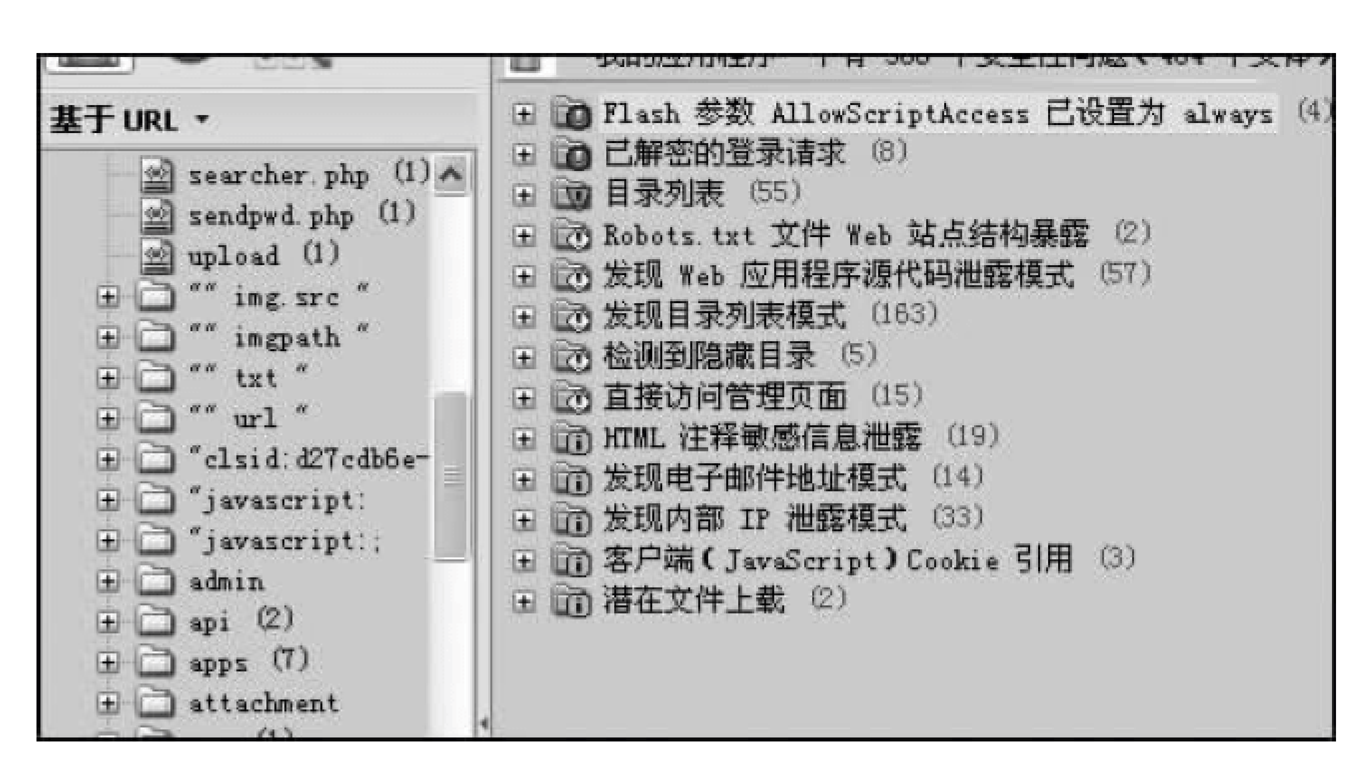


图 8.9　PHPWind 论坛漏洞

表 8.12　网站类别贡献值

网站类别	可信性级别	次序	贡献值
办公及政府机构网站、交易类网站、有偿资讯类网站	1	1	−1
企业品牌类网站、资讯门户类网站、功能性网站	2	2	−2
互动游戏网站、社区网站	3	3	−3

由于在本节模型中影响网站信任度的因素还包括网站的安全制度，但本节构建的两个网站均是在虚拟机中运行，未有任何安全制度，因此，本节采取相同的网站安全制度贡献值 0。

2. 计算并比较

在得到两个网站的贡献值后，要综合比较矩阵计算所得到的权重值和网站的类型贡献值，则可通过以下两个公式计算得到各个网站的信任度值：

$$t_j = \sum_{i=1}^{n} G_i (W_{C_{dj}})_i$$

$$T = \sum_{j=1}^{p} W_{z_i} t_j$$

动网论坛的信任度计算如下：

$$T_{po} = \sum_{j=1}^{p} W_{z_i} t_j = ((-9.67)+(-3.267)+(-0.1722))/3 = -4.3697$$

PHPWind 论坛的信任度计算如下：

$$T_{PH} = \sum_{j=1}^{p} W_{z_i} t_j = ((-4.264)+(-0.1722))/3 = -1.4787$$

由结果可知 PHPWind 论坛的信任度要高于动网论坛的信任度。

3. 结论

本节构建的动网论坛的版本比较早，且扫描软件获得的链接数仅为 73；而 PHPWind 论坛的版本为 8.7，为最新版，且其链接数为 1894。尽管相同时期的两种论坛版本的代码质量和网上评价近乎相同，但由于 PHPWind 论坛的版本较新，并且链接数量和代码量相对于动网论坛较多，因此 PHPWind 论坛的信任度要高于动网论坛的信任度。在本节中运用两种目前比较流行的网站进行比较并且在计算中得到了相同的结果，证明了本节模型的可用

性和时效性。

8.4 网络个人用户信任评价体系

本节提出网络个人用户信任评价体系的基础是网络实名制，只有实现了网络实名制，才能更好地将网络个人用户的网络身份和真实身份联系在一起。考虑到目前实行的实名制只有直接实名制和间接实名制，并且其在实行时还存在缺陷，因此，本节提出实行差别化实名制来区分不同的场合、不同的意愿和不同的人群的情况，即对于不同的场合、不同的网民意愿、不同的人群实行不同的实名制，并且提出了差别化实名制的实现方式——基于 PKI 的数字证书认证。在计算网络个人信任方面，本节提出通过综合运用层次分析法和模糊层次分析法来确定影响网络个人用户信任因素的权重，建议通过专家或专业人士综合相关方面来确定各个因素的信任贡献值，用以确定网络个人用户的信任。

8.4.1 差别化网络实名制

长期以来，开放性和匿名性是网络空间的显著特性，所谓匿名性是指网络用户（即网民）在互联网上经常以匿名的形式进行各种网络活动，网民一般以某种符号化的网名而不是真实姓名使用网络服务。可以说，这种匿名性在一定时期内促进了互联网的快速发展，但是也带来了网络空间下法律关系主体缺失，进而导致法律失灵的问题，由此引起诸如网络诚信缺失、网络侵权和犯罪等问题，同时也给网络个人用户的诚信度和信任评价工作带来困难。针对由于网络的匿名性产生的问题，各国政府越来越重视研究网络实名制。所谓网络实名制，是指在网络空间里参与主体的身份都是真实的主体身份，并且与现实中的真实身份对应。如果没有网络实名制，想要确定某些消息的散布者是谁或谁是某次攻击的操纵者都是比较困难的问题。由于无法确定个人用户的主体身份，就很难确定与之相关的信息、行为的真实性和可靠性。但是在实施网络实名制的同时，也可能对网络用户造成伤害，由于实名制要求网络 ID 与真实身份关联在一起，从而增加了暴露个人信息的风险，如人肉搜索，又如虐猫事件、死亡博客事件和辽宁女事件等，而“艳照门”的影响更是不言而喻。

1. *差别化实名制设计*

个人信息的搜集与整理可以为网络个人用户建立信用档案、进行信用评级提供数据来源与证据。实名制为搜集个人信息提供了体制保障和法律约束，但在人们争论实名制是否应该施行和是否会对网络活动起到积极作用时，他们对实名制还不是非常了解。目前专家学者关注的实名制分为直接实名制和间接实名制。间接实名制，又称后台实名制，是一种“后台实名、前台匿名”的实名制度。而直接实名制，顾名思义，就是前后台都采用实名制，通过用户的网络身份可知其真实身份。实名制本身可分为不同的类型并运用不同的方法，它是为了保护网络健康环境而建立的制度，它的制订与实施都是可以随环境和适用群体的不同而改变的。传统的直接实名制和间接实名制都存在不足，因此，本节提出建立一种差别化网络实名制，用以解决在对个人信息进行保护的同时成功地采集用户身份以及与其相关的信息的问题。差别化网络实名制有 3 层含义：第一是区分场合，第二是区分意愿，第三是区分人群。也就是说，对于不同的场合、不同的网民意愿和不同的人群实行不同的实名制。

2. 差别化实名制的实现

从网站自身拥有的技术来看,在操作层面上,网站可以做到要求网民在使用博客和论坛等交互式栏目时在后台使用实名注册,但允许网民在前台用匿名的方式登录。这样,登录者的实名对网站管理者来说是透明的,而对其他网民则是隐蔽的。网民之间只能相互看到各自的昵称,而不会看到各自的真实姓名及其他个人隐私。目前实现差别化实名制在技术上是成熟的,缺少的只是统一的、有效的认证制度和认证方法。

因此,本节提出通过采用基于 PKI 的数字证书来实现差别化网络实名制。即由国家来规范 CA 的建设,设计用户认证过程并且应用用户移动证书。通过建立国家级认证中心,统一颁发电子证书保证用户实体的可信任认证;通过用户认证过程来保证认证的可信性;通过移动证书来保证不同的场合、不同的意愿和不同的群体有不同的实名制限制;通过用户携带的 USBKey 作为认证媒介可以实现实时认证。这样使用户在不同的网站实名程度不同,在要求严格的网站完全实名,在可选实名时按用户意愿选择,通过实名验证的年龄登录不同的网站等。由于经过加密的个人信息存储于移动证书中,若要在不同的场合、对不同的用户意愿或针对不同的群体实现差别化的实名制,就需要在网站的服务端进行相关的配置。当用户想要登录直接实名制的网站时,该网站会提示以各种方式证明身份,在本节中以 USBKey 作为认证媒介,即插入 USBKey 移动证书。证书插入后,从客户端向网站服务器端发出认证请求,网站服务器端通过解密算法对客户端发出的加密后的身份信息进行验证。验证成功后,服务器端会向客户端发送验证成功消息,并提示是否适合浏览该网站。若此网站有不同的实名制板块时,验证成功后用户可以自主选择登录不同的板块。在整个认证的过程中身份信息都是以加密的形式传送的,并且服务器端、客户端和浏览器中不会保存任何身份信息。运用这种方法既可以确定网络个人用户的身份,同时又可保证用户身份及其相关信息不被泄露。

8.4.2 网络个人用户评价指标体系

网络个人用户是网络上最大的群体,他们用即时聊天软件聊天,用网络邮箱收发邮件,在各大论坛发帖,还有近几年比较流行的博客和微博,到处都有他们的身影。但若要采集可靠的、准确的网络个人用户信用数据,首先要有网络实名制作为基础;其次,采集数据的途径也非常重要。随着互联网的发展,人们的大部分日常生活都可以通过网络来实现,网络和人们的现实生活已经紧密地连接在一起。因此,在网络实名制的基础上,现实社会中的信用数据也是影响网络个人用户信任度的重要因素。当然网络个人用户的信任度还取决于其在网络上的行为,如聊天记录、邮件、网站浏览和下载记录、在论坛中发布的帖子、网络银行的交易记录,当然还包括在自己博客或微博中发布的消息和上传的图片等。这些数据中有些涉及个人的隐私,这些我们无权获取,但一旦个人用户通过各种方式进行有害个人信用的行为都会留下痕迹。可以通过很多方法获得这些数据,如即时聊天软件 QQ 的举报功能、论坛管理员查封的不合法帖子、各大交易网站中客户的交易记录、网络游戏的举报功能、网络上的举报中心(如网址为 www.12321.cn 的 12321 网络不良与垃圾信息举报受理中心,或网址为 net.china.com.cn 的中国互联网违法和不良信息举报中心等)、网络违法犯罪举报网站(www.cyberpolice.cn)以及各公司的防火墙得到的相关数据等。

因此,本节的网络个人用户评价指标体系不仅包括现实社会中的信用数据,还包括以上

提到的从网络中搜集到的相关数据。网络个人用户的评价指标体系如图 8.10 所示。

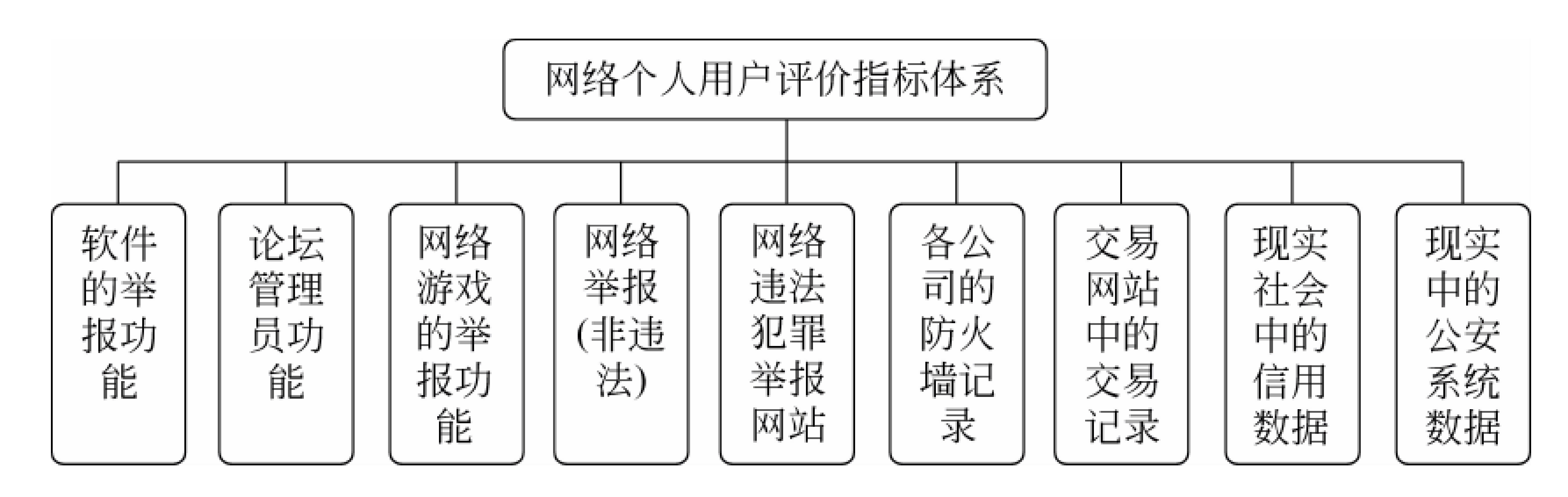


图 8.10　网络个人用户评价指标体系

由于涉及个人的隐私,要得到网络个人用户在网络上进行的所有操作的记录是不现实的。因此,本节建立的网络个人用户评价的指标体系中包括的因素并不是所有影响网络个人用户的因素,而是一些不涉及过多个人隐私的且相对比较容易得到的数据。

8.4.3　信任评价

网络个人用户的信用度评价不仅要考虑到技术、法律法规和人文因素,还需要大众的参与、各个部门的配合与社会各界的关注。这不仅仅是由于网络个人用户的信用度评价需要各个方面的、大量的数据,更是由于个人用户的信用的意义重大。个人的信用不仅是人与人交往的基础,更是构建和谐社会的基础和精神保障。个人的信用可以在网络交友、论坛发布消息、下载某人上传的资料、人才招聘甚至网上做交易时作为参考,同时在做这些事的时候自己的信用也会随之变化。个人的网络行为和网站不同,网站的行为可以在计算机和服务器中留有记录,而个人的行为是移动中的,可能在不同的地点出现并运用不同的软件。同时,属于智能生物的人类其行为可能受到的影响因素众多,从不同的环境到不同的心情,这些都是与软件和网站不同的。因此,网络个人用户的信用度评价的方法更加复杂,并且要加以考虑的因素更多。

1. 模型框架及方法

考虑到影响网络个人用户信任度的因素众多并且涉及不同的方面,可以采取结合多种方法的综合分级评价方式,因此,模型框架呈层次结构,如图 8.11 所示。

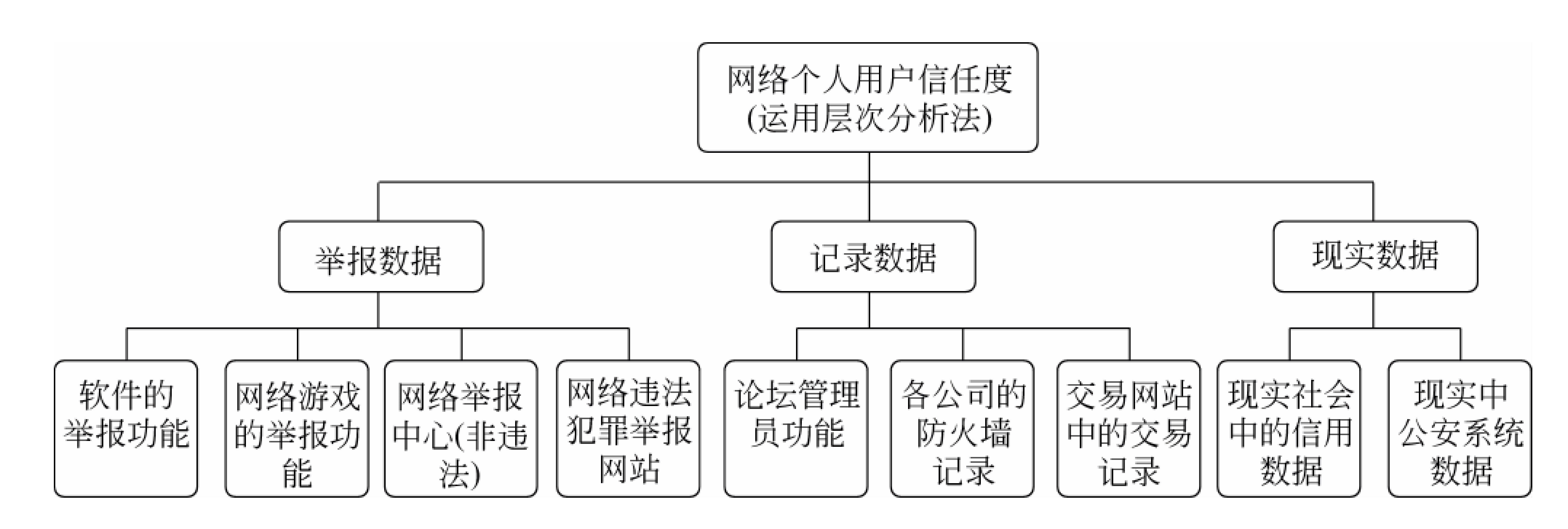


图 8.11　网络个人用户信任度分类指标体系

由于准则层中的数据的可信性很明显,但在子准则层中可能得到的数据会有很大的不确定性,因此,本节采用对准则层运用 AHP 方法,对于举报数据的子准则层运用模糊层次

分析法(Fuzzy AHP,FAHP)来确定举报数据中各个因素的权重值。

2. 构造比较矩阵

由于数据的来源不同其本身的信任度也有不同,来自举报功能的举报数据的信任性要比管理员功能和现实数据差;而从管理员、防火墙和交易记录中的数据的信任性要高于举报数据但低于现实中的数据。因此,准则层比较矩阵 $\boldsymbol{F}$ 为

$$\boldsymbol{F}=\begin{bmatrix}1 & 2 & 3\\ \frac{1}{2} & 1 & 2\\ \frac{1}{3} & \frac{1}{2} & 1\end{bmatrix}$$

对于举报数据子准则层中的数据运用模糊层次分析法。

此种方法在构造两两比较判断矩阵时,用模糊数代替确定值来表示所得到数据的模糊信息,将两两比较值模糊化从而得到模糊权重,最后通过非模糊化将模糊判断的不确定性在形式上转换为确定性[1~3]。

定义 8.1 设 X 为论域,$u_{\widetilde{M}}(x)$表示 X 到闭区间[0,1]上的函数,称 $u_{\widetilde{M}}(x)$确定了一个 X 上的模糊集合 $\widetilde{M}$,记 $\widetilde{M}=\{u_{\widetilde{M}}(x)\mid x\in X\}$,并称 $u_{\widetilde{M}}(x)$为对 $\widetilde{M}$ 的隶属函数。对确定的元素 $x_0\in X$,函数值 $u_{\widetilde{M}}(x_0)$称为元素 x_0 对于模糊集合 $\widetilde{M}$ 的隶属度。如果 $u_{\widetilde{M}}(x)=0.7$,表示元素 x 以 0. 8 的隶属度属于模糊集合 $\widetilde{M}$,而以 0. 4 的程度不属于集合 $u_{\widetilde{M}}$。因此,可通过调查得到的数据从而得到举报子准则层的比较矩阵 $\boldsymbol{G}$:

$$\boldsymbol{G}=\begin{bmatrix}1 & \widetilde{a}_{12} & \widetilde{a}_{13} & \widetilde{a}_{14}\\ \widetilde{a}_{21} & 1 & \widetilde{a}_{23} & \widetilde{a}_{24}\\ \widetilde{a}_{31} & \widetilde{a}_{32} & 1 & \widetilde{a}_{34}\\ \widetilde{a}_{41} & \widetilde{a}_{42} & \widetilde{a}_{43} & 1\end{bmatrix}$$

记录数据中可以得到的数据是确定的数据,但是从获得途径可以看出,交易网站中的交易记录是信任度最高的,因为这些交易记录都是公开给用户看的。之所以选取交易记录,是因为一些网络个人用户可能在网上购物的时候进行行骗或给卖家恶意评价来牟取利益。同时,从论坛管理员功能和公司的防火墙得到的记录数据都要经过管理员和公司的处理,处理结果可能只是记录数据的一部分,因为其中可能混杂了该公司的一些私密信息。因此,对于记录数据子准则层的比较矩阵,可通过专家数据或调查问卷得到相关比较矩阵 $\boldsymbol{H}$:

$$\boldsymbol{H}=\begin{bmatrix}1 & a_{12} & a_{13}\\ a_{21} & 1 & a_{23}\\ a_{31} & a_{32} & 1\end{bmatrix}$$

对于现实数据来说,本节中列出的两种影响因素重要性相同,所以比较矩阵 $\boldsymbol{R}$ 为单位阵。

$$\boldsymbol{R}=\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}$$

3. 信任度贡献值的确定与计算

通过对矩阵进行每列归一化、矩阵的各元素按行相加、向量的归一化并进行一致性检验

后，应用通过一致性检验的矩阵或经过调整的矩阵进行计算，得出权重值用于评价。

网络个人用户的信任度会被作为用户进行网络操作行为时的参考，其意义重大，所以各个子准则层和准则层元素的贡献值的确定，就需要由专家或专业人士综合相关法律法规、情景情况以及人文宗教等各个方面的因素来进行，以得出个人的信任度贡献值，最后通过各个因素的贡献值和权重计算网络个人用户的信任度。

8.5　本章小结

为了加强网络诚信建设，必须建立一套有效的网络主体信任评价标准、评价方法和评价体系。本章主要探讨了 3 种网络主体评价体系：软件信任评价体系、网站信任评价体系和网络个人用户信任评价体系。首先，采用 AHP 方法建立软件信任评价模型，分析了影响软件信任度的因素，并运用调查问卷的形式得到相关的软件评价数据，用以对软件评价模型进行检验。然后，基于 AHP 方法对网站的信任评价进行建模，模型中不仅考虑到比较受关注的网站漏洞，还考虑了网站类别和网站安全制度等因素，并通过应用虚拟机技术建立两个不同的网站验证了模型的可用性。最后，在讨论网络个人用户信任评价体系的过程中，提出了一种差别化网络实名制，探讨了这种差别化网络实名制的实现方法，在此基础上详细分析了网络个人用户信任评价的指标体系和信任评价模型。

参 考 文 献

[1]　吴诗佑. 网络诚信建设研究[D]. 电子科技大学硕士论文，2005.

[2]　吴其聪，顾健，陆臻. 数字证书技术在网络实名制中的应用研究[J]. 技术研究，2011，6：91-96.

[3]　卢涛，董坚峰. 中美电子商务网站评价比较研究[J]. 情报科学，2008，26(4)：591-594.

[4]　姜琳. 美国 FICO 评分系统述评[J]. 商业研究，2006，20：81-84.

[5]　李伟舵. 国外征信体系建设的基本经验[J]. 他山之石，2006，33：306.

[6]　Kang Shao，Yuanjing Guo，Chaohua Liu. The Model of Credit Evaluation in C2C E-commerce[A]. In：Information Management and Engineering (ICIME)，2010 The 2nd IEEE International Conference [C]. Chengdu：IEEE，2010：118.

[7]　Haitao Su，Haiqing Guo. C2C E-Business Seller Credit Evaluation Model Based on Dynamic Weighting[A]. In：Management and Service Science (MASS)，2010 International Conference[C]. Wuhan：IEEE，2010. 1.

[8]　吴洋. 网络主体信任评价体系研究，北京信息科技大学硕士学位论文，2011. 12.